UNIVERSIDAD AUTONÓMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Análisis, comparación y predicción de interacciones de usuarios en cursos online

Máster universitario en Ingeniería Informática

Autor: GONZÁLEZ-GALLEGO SOSA, Miguel Ángel

Tutor: PULIDO CAÑABATE, Estrella

FECHA: Febrero, 2018

Agradecimientos

En primer lugar me gustaría agradecer a Estrella y Gonzalo por toda la implicación docente que han tenido conmigo a lo largo de estos últimos años de universidad y lo mucho que he aprendido gracias a ellos. Por otro lado a mi compañero Ángel por la ayuda y la motivación que me ha brindado desde que le conozco. Además agradecer enormemente a la Cátedra UAM/IBM que me ha proporcionado una beca con la cual me he estado formando continuamente y no he dejado de aprender en ningún momento. Finalmente quiero agradecer a mi familia y amigos por el apoyo recibido durante mi paso por esta etapa universitaria, sin ellos nada de esto hubiese sido posible.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 Introducción

Los MOOCs [1] son cursos en línea masivos que están cobrando una gran relevancia en el ámbito de la educación a través de internet. La educación online es de gran utilidad, debido a que se crea una conexión a distancia entre profesores y estudiantes a través de las nuevas tecnologías haciendo uso de las facilidades que proporciona internet. Las ventajas en este ámbito son amplias por ejemplo se dispone de gran cantidad de posibilidades de formación, se permite seguir los cursos desde cualquier lugar, te ofrecen la posibilidad de avanzar a tu propio ritmo etc.

Los MOOCs son impartidos por equipos de profesores universitarios u otras instituciones docentes, por ejemplo, profesores de la Universidad Autónoma de Madrid se encuentran actualmente en la plataforma *edX* [2] impartiendo cursos online. Estos profesores son los encargados de compartir el material con el cual se impartirá el curso, vídeos, documentos, problemas etc. Todo esto es utilizado por los estudiantes de manera online, y sus interacciones quedan registradas en los logs de la plataforma en la que se imparte el curso.

En este contexto el objetivo de este TFM es la creación de un sistema que permita el análisis de las interacciones realizadas por los estudiantes en cualquier curso online ofertado en la plataforma *edX* utilizando su patrón de acceso a la plataforma. Además, se emplearán algoritmos de aprendizaje automático para predecir variables de interés sobre la interacción de los estudiantes con los cursos. Una de ellas es la tasa de abandono que es muy alta en este tipo de cursos. Respecto al rendimiento académico de los estudiantes se pretende crear un modelo que permita predecir la calificación de los estudiantes. Los resultados de estas predicciones se compararán con diferentes ediciones de cursos online.

1.1 Objetivos

Los objetivos específicos de este trabajo son:

- Extracción de eventos para diferentes cursos MOOC. Se pretende realizar un programa que facilite la extracción de eventos para cursos MOOCs independientemente de los tipos de actividades que estos contengan.

- Automatización del flujo de extracción de características para MOOCs. En esta tarea se pretende realizar un programa que lleve a cabo una extracción de características automática a partir de los eventos realizados por los usuarios de cursos MOOC. Con este objetivo se podrán crear numerosos atributos que nos permitirán predecir variables de interés de los estudiantes en el curso. Este programa se debe dotar de la mayor flexibilidad posible para que, independientemente del curso que estemos analizando, la extracción de eventos sea sencilla, es decir, sin que sea necesario modificar numerosos parámetros o crear nuevas funciones para analizar un curso u otro.

- Realización de una comparación de los resultados obtenidos a partir de diferentes ediciones de cursos online, con la finalidad de extraer información útil y relevante que permita mejorar los cursos disminuyendo la tasa de abandono y aumentando el rendimiento.

1.2 Competencias adquiridas

Para la realización del trabajo se precisa del conocimiento de las siguientes técnicas, herramientas y lenguajes:

- *Apache Spark*: herramienta para el manejo de grandes volúmenes de datos.
- *Python*: lenguaje de programación interpretado que se usa en nuestro análisis.
- *Dataframes*: estructura de datos similar a las tablas SQL y proporcionada por la librería pandas de Python.
- *RDD*: conjuntos de datos distribuidos sobre los que trabaja Spark.
- *Sklearn*: librería para aprendizaje automático de Python.
- *Random Forest, regresión logística y XGBoost*: algoritmos de aprendizaje automático para llevar a cabo la predicción de abandono y la calificación del examen final.
- *Curvas ROC*: métrica para la evaluación de los resultados obtenidos en la predicción de abandono.

Estas son algunas de las competencias adquiridas más importantes y que han sido necesarias a la hora de realizar el trabajo.

1.3 Organización de la memoria

La memoria está dividida en los siguientes capítulos:

- *Estado del arte*: en este capítulo se expone el contexto del trabajo en el ámbito de la educación a través de internet.
- *Preprocesado de datos con Spark*: en este capítulo se describen las herramientas utilizadas y el proceso de preprocesado de datos mediante la herramienta Spark. Además, se explican los eventos de los cursos online, el formato de los logs de edX y se describe como se realiza el preprocesado para diferentes ediciones.
- *Automatización del flujo de datos:* en este capítulo se describe cómo se lleva a cabo el programa que realiza el flujo de datos de manera flexible para cualquier curso online. En esta automatización se realiza el proceso de extracción de características de los estudiantes al igual que el de la extracción de las etiquetas para la predicción del abandono y la calificación del examen final.
- *Aplicación de técnicas de aprendizaje:* en este capítulo se describen:
    - Los algoritmos de aprendizaje automático que se utilizarán para llevar a cabo la predicción del abandono y la calificación del examen final.
    - Los datos obtenidos tras la automatización, es decir los datos que se utilizarán en la predicción.

- El método de aprendizaje automático para la predicción del abandono y de la calificación de la nota del examen final de los estudiantes del curso online.

- Los resultados obtenidos para las diferentes ediciones del curso online con la finalidad de extraer información útil sobre los estudiantes y el curso.

2 Estado del arte

En los últimos años numerosos estudios de investigación han utilizado los datos generados por las plataformas de enseñanza online para obtener información sobre el comportamiento, el rendimiento y los resultados de los estudiantes, la eficacia de los cursos para mantener a los usuarios activos en la plataforma y la transmisión de conocimientos. A continuación repasaremos algunos de estos estudios, empezando por los más generales y centrándonos en aquellos que se relacionan con el abandono de un curso, analizando la forma en que se manejan los diferentes criterios y modelos de predicción.

A diferencia de la docencia tradicional, los estudiantes matriculados en los MOOCs a menudo muestran una gran variedad de motivaciones y niveles de compromiso con las actividades propuestas y el contenido ofrecido en el curso: algunos quieren obtener un certificado, otros desean actualizarse en el conocimiento de un tema en particular o simplemente ver de qué trata el curso [3]. Como resultado, los MOOC muestran tasas de abandono muy altas, lo que ha causado tanto la preocupación de sus promotores como la curiosidad de los investigadores [4]**.** La pregunta es ¿cómo podemos abordar estos factores para tratar de minimizar este abandono?

A partir de la literatura centrada en Learning Analytics y Machine Learning, los investigadores han tratado de encontrar soluciones para enfrentar los desafíos que surgieron de las plataformas de enseñanza online [5] [6] La mayoría de los estudios coinciden en destacar el abandono como uno de los grandes desafíos para este tipo de técnicas analíticas. De hecho, el problema del abandono en los MOOC ha ido más allá, del mundo académico a una de las competiciones de aprendizaje automático más prestigiosas del mundo: la KDD Cup en su edición de 2015, donde los competidores intentaron predecir el abandono utilizando diferentes características sobre estudiantes y cursos.

Volviendo a la parte académica, algunos investigadores trataron de comprender el problema del abandono utilizando la agrupación k-means en 28 MOOCs. Identificaron cinco grupos diferentes de escenarios: estudiantes que solo se inscriben, estudiantes con poco compromiso, estudiantes que únicamente interaccionan con videos, estudiantes que interaccionan con videos y actividades, y estudiantes que utilizan el foro [7]**.** Por simplicidad, en general, hay dos grandes grupos de causas que explican este problema de abandono. Por un lado, como hemos mencionado anteriormente, las diferentes motivaciones de los estudiantes, que pueden conducir desde la frustración al aburrimiento [8] y, por el otro, el diseño, la presentación y la calidad de los cursos teniendo en cuenta el perfil del alumno [9].

El primer aspecto que se debe aclarar al llevar a cabo el análisis de abandono es cómo lo definimos. No existe una definición universalmente aceptada de abandono en los MOOC, y diferentes investigadores y analistas han utilizado sus propias definiciones de acuerdo con sus estudios, por lo que las comparaciones son difíciles de realizar. Por el momento, consideraremos dos definiciones que agrupan diferentes formulaciones de abandono de un alumno en un curso. Cada una de estas definiciones aporta diferentes matices según el objetivo de los investigadores que las utilizan.

- **Definición 1**: se considera que un estudiante ha abandonado un curso si no ha participado en las actividades de la última semana del curso [10] [5] [6]. La

principal debilidad de este enfoque es que no captura la actividad de los estudiantes a lo largo del curso y puede ser sensible a los usuarios que, habiendo alcanzado el objetivo en la calificación, no completan las actividades de la última semana.

- **Definición 2**: Dentro del análisis de abandono en una perspectiva temporal, nos referimos al abandono del incumplimiento de actividades en la semana actual. Supone que no aparece ningún evento durante ningún día de la semana que se analiza. [11] [12] [13] [14] [15].

Teniendo en cuenta estas definiciones, para este trabajo se han establecido dos tipos de criterios de abandono en función de las interacciones de los estudiantes:

- **Criterio 1:** en este criterio consideraremos que un estudiante ha abandonado si no ha realizado ninguna actividad evaluable (problemas o proyectos) en las dos semanas siguientes a la considerada. Este criterio está relacionado con las dos definiciones anteriores ya que en la *definición 2* utilizamos la información temporal de las semanas y en la *definición 1* la información de las actividades evaluables. Implementaremos este nuevo criterio ya que puede ocurrir que un estudiante no realice ninguna actividad evaluable en una semana y continúe en el curso, pero por el contrario si en dos semanas no hace actividades evaluables es más probable que ese estudiante haya abandonado.

- **Criterio 2:** en este criterio consideramos que un estudiante ha abandonado si no ha realizado ningún evento en la siguiente semana a la considerada. Este criterio es similar a la *definición 2* descrita anteriormente pero en vez de tener en cuenta solo los problemas, consideramos que un estudiante ha abandonado si no ha realizado ningún evento, no solo las actividades como aparece en la *definición 2*. Implementaremos este nuevo criterio ya que si un estudiante no hace nada durante una semana se habrá perdido la continuidad en el curso y seguramente haya abandonado.

Otro aspecto relevante es el tipo de evento que evaluaremos a la hora de considerar que el usuario ha abandonado el curso, para tratar de no descartar a los usuarios que, por ejemplo, regresan al curso puntualmente para realizar una acción diferente a las que se han considerado.

También es necesario considerar el conjunto de atributos o características que se analizarán para clasificar a los estudiantes entre los que abandonan y los que no. Lo más común ha sido evaluar los diferentes atributos de los estudiantes, tanto socioeconómicos como de las actividades realizadas durante el curso.

Algunos autores han abordado el tema del abandono teniendo en cuenta no solo la información (agregada y temporal) de las semanas previas del curso analizado (lo que denominan aprendizaje in situ), sino también otros aspectos, como los patrones transferidos de otros cursos, para tratar de mejorar el algoritmo [16] [17].

Otros autores analizan el comportamiento y el rendimiento en el MOOC estudiando los patrones de interacción con los videos de los cursos: por ejemplo, a partir de la secuencia de eventos creados, y las posiciones visitadas en un video [18]. Con el análisis basado en eventos, se extraen las características fundamentales recurrentes (como secuencias de reproducción y pausa) y las secuencias de posición (desplazamiento de video) que pueden

indicar las dificultades del usuario para comprender el contenido de los videos. Los modelos predictivos que utilizan estos patrones pueden mejorar sustancialmente la calidad de la predicción en términos de precisión. Según los autores, estos modelos son útiles en situaciones donde los datos de entrenamiento son limitados, como la detección temprana en las primeras semanas o los cursos cortos [19]. Algunos autores también intentan predecir la calificación final para los alumnos antes de los exámenes finales [20].

Otros autores analizan los MOOC utilizando principios de la microeconomía. Usando el modelo de riesgos proporcionales de Cox, analizan la tasa de desgaste e incluyen datos demográficos, encontrando que los estudiantes más jóvenes, los participantes de EE. UU. y las mujeres tienen menos probabilidades de completar los cursos [21].

También existen estudios más concretos de aspectos particulares de los cursos, como el análisis de la participación en los foros, el uso de técnicas de gamificación para aumentar la participación y la comunicación profesor-alumno y alumno-alumno [22], el análisis de sentimiento a través de los comentarios de los foros para comprender mejor la interacción de los usuarios con la plataforma [23], la predicción de la participación de los estudiantes en las actividades de revisión por pares [24] o el uso de hashtags en Twitter con la técnica de análisis de redes sociales (SNA) para descubrir centros y personas influyentes en la red [25].

En resumen, existen una gran variedad de perspectivas y modelos para analizar los MOOC. En general, no hay ninguno que pueda generalizarse debido a la diversidad de formatos de los cursos. Y esta variedad va en aumento, ya que las plataformas han optado por diferentes estrategias en su evolución: algunas ofrecen micromasters, otras cursos orientados a la formación práctica y, sobre todo, cursos a su propio ritmo. Como veremos más adelante, el modelo para el que apostaremos también depende del tipo de curso que estamos analizando, pero esperamos que a lo largo del proyecto en el que se incluye este trabajo, se pueda obtener un modelo más generalizable. El objetivo no es solo obtener un buen modelo predictivo de abandono y rendimiento, sino también extraer valor de los datos para comprender mejor a los estudiantes y sus necesidades y proporcionar recursos y actividades útiles y estimulantes.

Como hemos podido observar existen diversos métodos de análisis de cursos online. En este trabajo se pretende crear un programa que se encargue del preprocesado de los datos de cualquier curso online mediante una herramienta para el manejo de grandes volúmenes de datos. Además de este preprocesado, automatizaremos el flujo de los datos permitiendo la extracción de nuevas características de manera sencilla para cualquier curso online de la plataforma edX. Por último, se aplicarán técnicas de aprendizaje automático para la predicción de variables de interés de los cursos de manera automática e independiente del curso online que se analice. Tras explorar en numerosas fuentes no tenemos constancia de la existencia de este tipo de programas por lo que consideramos que esto es innovador a la hora de aplicarse en temas de *Learning Analytics*.

3 Preprocesado de datos con Spark

En este apartado se describe la tarea del preprocesado de datos para cursos online de la plataforma edX. Para ello se emplea Spark que, como hemos mencionado anteriormente, es una herramienta para el manejo de grandes volúmenes de datos. El objetivo principal de esta tarea es crear un programa que requiera las modificaciones mínimas para extraer la información relevante de los logs de edX, para distintas ediciones de cursos online, en un formato más sencillo de procesar.

3.1 Spark

Apache Spark es una herramienta de computación en paralelo para el procesamiento de datos masivos. Es el sucesor de Hadoop, otra herramienta para el procesamiento de grandes volúmenes de datos, basado también en las funciones de mapeo y reducción. Sin embargo, Spark es más rápido ya que trabaja en memoria. Una comparativa de la velocidad del algoritmo de regresión logística en ambos sistemas de muestra en la Figura 1 [26].

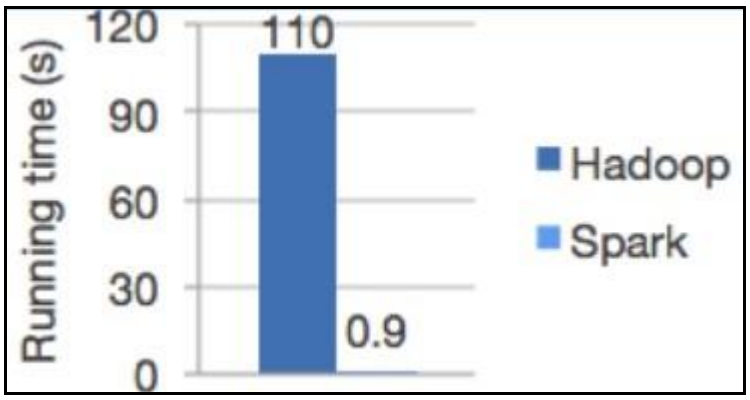

Figura 1. Spark vs Hadoop (Imagen extraída de [26])

Además de esta comparativa de velocidad, Spark mejora a Hadoop en múltiples aspectos, por ejemplo: la simplicidad del código, la disposición y simplicidad de varias API's (Python, Java, Scala) y la integración de distintas librerías para la creación de diversas aplicaciones.

3.1.1 Arquitectura de Spark

La arquitectura de Spark está compuesta principalmente por los siguientes componentes:

- *SparkContext*: variable de entorno de Apache Spark.
- *Programa driver*: proceso que ejecuta la función *main()* de la aplicación y la creación del SparkContext
- *Cluster Manager*: servicio externo para la adquisición de recursos en el clúster.
- *Worker node*: cualquier nodo que pueda ejecutar código en el clúster.
- *Ejecutores*: proceso iniciado por una aplicación en los *worker nodes*. Este proceso ejecuta *tareas* y mantiene los datos en memoria o en almacenamiento en disco.

- *Tareas:* porciones de código de la que se encargan los ejecutores.

Las aplicaciones Spark se ejecutan como conjuntos de procesos independientes en un clúster. Estos procesos estan coordinados por el objeto *SparkContext* en su programa principal llamado *driver*. Esta variable se conecta a los gestores de clúster que son los encargados de asignar recursos al sistema. A partir de esa conexión se crean los *ejecutores* que permiten que se compute en los *worker node*. Finalmente en esos *ejecutores* se encuentran las porciones de código denominadas *tareas*. Esta estructura de Spark se muestra en la Figura 2 [27].

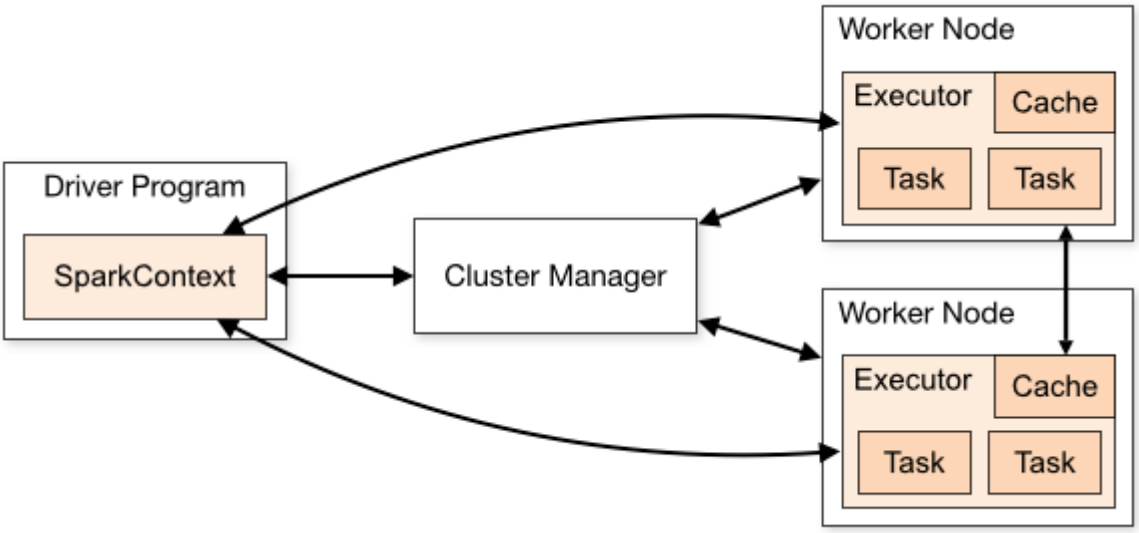

Figura 2. Arquitectura Apache Spark

Además, Spark proporciona API's de alto nivel para Java, Python y Scala e incluye una serie de librerías específicas para poder llevar a cabo aplicaciones más especializadas (véase la Figura 3):

- *Spark SQL:* para el tratamiento de datos estructurados.
- *Spark Streaming:* para el procesamiento de datos en tiempo real.
- *MLlib:* para llevar a cabo tareas de aprendizaje automático.
- *Graphx:* para la computación sobre grafos.

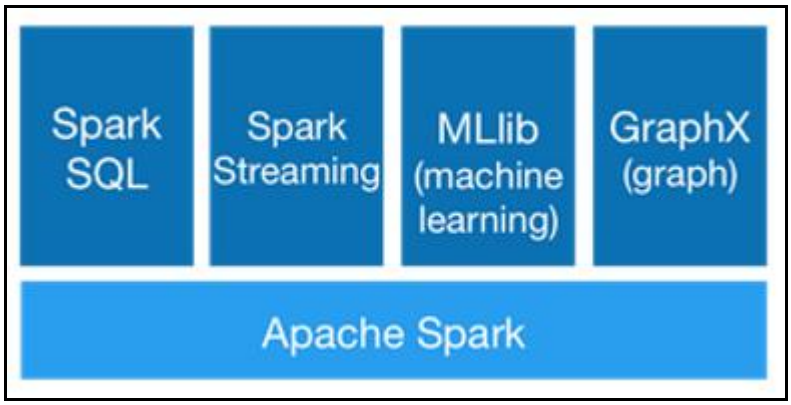

Figura 3. Librerias Spark

En nuestra aplicación no se usará ninguna de las librerías mencionadas anteriormente. Simplemente se utiliza la tecnología proporcionada por Apache Spark básico basada en colecciones de datos denominadas RDDs y que explicaremos a continuación.

3.1.1 RDD (Resilient Distributed Dataset)

En Spark se trabaja sobre colecciones de datos denominadas RDDs. Estas colecciones se caracterizan porque las operaciones realizadas sobre ellas se ejecutan en paralelo. Las principales características de los RDDs son las siguientes:

- Son inmutables. Es decir, se pueden aplicar transformaciones para crear nuevos RDDs o realizar acciones sobre ellos pero no modificarlos.

- Se guarda la secuencia de transformaciones para poder recuperar RDDs de forma eficiente si alguna máquina falla, es decir son robustos a fallos del cluster (*Resilient*).

- Están distribuidos en los *nodos workers*.

El procesamiento de datos basados en RDD se realiza mediante dos tipos de operaciones: acciones y transformaciones. Las transformaciones son operaciones que devuelven otro RDD mientras que las acciones son aquellas operaciones que devuelven un resultado al *driver*. Un programa Spark generalmente está formado por una serie de transformaciones y una o varias acciones. Las transformaciones no se ejecutan hasta que no se realiza una acción, este tipo de ejecución se denomina *"lazy evaluation"* o *"evaluación perezosa"*.

En la Figura 4 aparece un programa sencillo de Spark para contar el número de ocurrencias de cada palabra en un fichero de texto.

```
text_file = sc.textFile("prueba.txt")

counts = text_file.flatMap(lambda line: line.split(" "))
                        .map(lambda word: (word, 1))
                        .reduceByKey(lambda a, b: a + b)

print(counts.collect())
```

Figura 4. Contar palabras Spark

En la primera línea del programa mediante la variable de entorno *sc*, utiliza la función *textFile* para cargar un fichero de texto. Una vez cargamos el fichero dispondremos automáticamente de cada línea del fichero en un único RDD, es decir, similar a una colección de *strings*. A continuación utilizaremos la transformación *flatmap*, que se encarga de dividir el texto en palabras mediante la función *Split*(“ ”). Es decir tendremos una lista de palabras en un RDD. A continuación, la función *map* asociará un 1 dentro del RDD a cada palabra. Llegados a este punto aún no se ha ejecutado nada ya que únicamente hemos usado transformaciones. A continuación se aplica la transformación *reduceByKey* que ejecutará la suma de los valores “1” anteriores por palabra (Key). Finalmente tendremos un RDD formado por cada palabra y su número de apariciones en el texto. Con la acción *collect*() ejecutaremos todas las transformaciones y todos los elementos del RDD obteniendo así el mapeado de cada palabra con su número de apariciones en el documento.

Si comparamos con el programa de contar palabras implementado en la tecnología Hadoop (Figura 5 [28]) podemos observar que Spark ofrece una mayor facilidad de uso y mejora respecto al número de líneas utilizadas gracias a la tecnología de los RDDs.

```
1  package org.myorg;
2
3  import java.io.IOException;
4  import java.util.*;
5
6  import org.apache.hadoop.fs.Path;
7  import org.apache.hadoop.conf.*;
8  import org.apache.hadoop.io.*;
9  import org.apache.hadoop.mapreduce.*;
10 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
11 import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
12 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
13 import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
14
15 public class WordCount {
16
17  public static class Map extends Mapper<LongWritable, Text, Text, IntWritable
18     private final static IntWritable one = new IntWritable(1);
19     private Text word = new Text();
20
21     public void map(LongWritable key, Text value, Context context) throws IOE
22         String line = value.toString();
23         StringTokenizer tokenizer = new StringTokenizer(line);
24         while (tokenizer.hasMoreTokens()) {
25             word.set(tokenizer.nextToken());
26             context.write(word, one);
27         }
28     }
29  }
30
31  public static class Reduce extends Reducer<Text, IntWritable, Text, IntWrita
32
33     public void reduce(Text key, Iterable<IntWritable> values, Context contex
34       throws IOException, InterruptedException {
35         int sum = 0;
36         for (IntWritable val : values) {
37             sum += val.get();
38         }
39         context.write(key, new IntWritable(sum));
40     }
41  }
42
43  public static void main(String[] args) throws Exception {
44     Configuration conf = new Configuration();
45
46         Job job = new Job(conf, "wordcount");
47
48     job.setOutputKeyClass(Text.class);
49     job.setOutputValueClass(IntWritable.class);
50
51     job.setMapperClass(Map.class);
52     job.setReducerClass(Reduce.class);
53
54     job.setInputFormatClass(TextInputFormat.class);
55     job.setOutputFormatClass(TextOutputFormat.class);
56
57     FileInputFormat.addInputPath(job, new Path(args[0]));
58     FileOutputFormat.setOutputPath(job, new Path(args[1]));
59
60     job.waitForCompletion(true);
61  }
62
63 }
```

Figura 5.Contar palabras Hadoop

3.1.2 Pyspark

Para poder llevar a cabo la tarea del preprocesado de datos se utiliza el intérprete interactivo de Python con Apache Spark llamado Pyspark. Para emplear este intérprete se instala la distribución de Python Anaconda, que junto con Jupyter Notebook, un entorno interactivo web de ejecución de código, nos permitirá realizar el preprocesado de datos.

3.2 Formato logs edX

La plataforma de cursos online edX registra las interacciones de los usuarios con la plataforma mediante logs en formato JSON (véase Figura 6). Cada interacción de los usuarios queda guardada en un registro del log con numerosa información. Para modelar el comportamiento de los estudiantes en cursos online se realiza el filtrado de eventos, ya que no todos los eventos ni su información asociada son útiles para modelar dicho comportamiento.


```
{"username": "", "event_source": "browser", "name": "pause_video", "accept_language":
"es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3", "time": "2015-10-05T01:17:59.551233+00:00", "event": "{\"code\":
\"uRq2UY83V7o\", \"id\": \"i4x-UAMx-Android301x-video-af51f346e9434851b56e27080a3bd6de\", \"currentTime\":
413.14}", "agent": "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:40.0 Gecko/20100101 Firefox/40.0", "label": "UAMx/
Android301x/1T2015", "host": "courses.edx.org", "session": "cebaa015fbc6f65fbcf6171484c0e5f7", "referer":
"https://courses.edx.org/courses/UAMx/Android301x/1T2015/
courseware/60016318f08a4e8cb77726eb2570b0da/3123f3bb280a4f2991e0f473a907a5ee/", "context": {"user_id":
7114428, "org_id": "UAMx", "course_id": "UAMx/Android301x/1T2015", "path": "/event"}, "ip":
"41e937f4deeb6d62732dcf824628898e", "page": "https://courses.edx.org/courses/UAMx/Android301x/1T2015/
courseware/60016318f08a4e8cb77726eb2570b0da/3123f3bb280a4f2991e0f473a907a5ee/", "nonInteraction": 1,
"event_type": "pause_video"}
```


Figura 6. Evento

Puede observarse que no toda la información del evento es relevante para el presente estudio. Un ejemplo es el campo org_id, que contiene el id de la organización que imparte el curso y no nos proporciona relevante. Para cualquier evento, independientemente del tipo que sea, siempre extraeremos los siguientes campos del log:

- *Event_type* : este campo del log contiene el tipo de evento, en este caso se trata de un evento de tipo play_video.
- *User_id*: contiene el id del usuario y se encuentra dentro del campo context del evento.
- *Time*: aquí se encuentra el timestamp en el que el evento se ha realizado.

Otro dato a tener en cuenta es que los eventos no se encuentran ordenados por tiempo de creación (*timestamp*) y por lo tanto es necesario ordenarlos una vez filtrados.

Para este trabajo, de todos los eventos generados por la plataforma edX, se han considerado los siguientes como los que describen mejor la interacción del usuario con el curso:

- **Documentos:** este tipo de eventos se generan cuando el estudiante interacciona con algún documento pdf del curso. Como información útil de este evento, guardaremos únicamente el id del documento así como la información mencionada anteriormente que poseen todos los eventos.

- **Vídeos:** estos eventos se generan cuando el estudiante interacciona con los vídeos del curso. De todos los eventos de vídeo, se han contemplado los siguientes:

    - *Pausar vídeo:* se desencadena cuando se pausa un vídeo y recogemos como información del evento el tiempo del vídeo en el cual se ha realizado la pausa por el estudiante.

    - *Play vídeo:* se desencadena cuando el estudiante empieza o reanuda un vídeo y recogemos como información útil del evento el tiempo del vídeo en el cual el estudiante ha realizado el evento.

    - *Cambio de velocidad en el vídeo:* se desencadena cuando un estudiante cambia la velocidad del vídeo y se registra la velocidad nueva y la antigua una vez filtrado el evento.

    - *Desplazamiento en el vídeo:* se desencadena cuando el estudiante se desplaza en el vídeo, y se registra la posición inicial y la posición final una vez filtrado el evento.

  Además, para todos estos eventos de vídeo quedarán registrados tanto los campos mencionados anteriormente (user_id, event_type, timestamp) como el id del vídeo al que hacen referencia.

- **Foro:** estos eventos se desencadenan cuando el estudiante interacciona con el foro del curso. Dentro de estos eventos hemos tenido en cuenta la creación de hilos, las respuestas a hilos, las respuestas a comentarios y las búsquedas en el foro. Para las tres primeras interacciones se registra: el texto del usuario, si el estudiante sigue el mensaje, el id del hilo, el id del foro del curso, y el id del mensaje al que se responde en caso de que se trate de una respuesta. En el caso que se trate de una búsqueda en el foro se registra el texto escrito por el estudiante para realizar la búsqueda.

- **Autoevaluación:** estos eventos se registran en el log cuando el estudiante realiza una autoevaluación sobre un proyecto o actividad del curso. Para estos eventos guardamos como información útil el id de la autoevaluación y las partes de las que está formada la autoevaluación. Además, para cada parte de la autoevaluación, se registra la máxima puntuación posible, la nota que se asigna el estudiante y el feedback del estudiante.

- **Problemas:** estos eventos se registran en el log cuando el estudiante hace un problema del curso. Un problema consta de uno o más ejercicios. Para cada problema guardamos el id del problema, el número de ejercicios que tiene, y el número de intentos que ha realizado el estudiante sobre ese problema. Además, como los problemas están formados por ejercicios, registraremos si el ejercicio se ha resuelto correctamente o no.

3.3 Preprocesado de los logs

Una vez identificados los eventos y campos relevantes para los cursos online, se ha desarrollado el programa basado en programación funcional que se encarga de generar un único fichero JSON mediante la tecnología de Apache Spark.

Para llevar a cabo este preprocesado de datos se han seguido los siguientes pasos: en primer lugar, se han cargado los datos de los distintos logs de un curso, seguidamente se han filtrado los datos en función de los eventos mencionados anteriormente y se ha extraído la información útil de cada uno de ellos. Por último, se han agrupado todos los eventos por estudiante y se han ordenado cronológicamente guardando el resultado en un fichero en formato JSON. Este fichero consta de una serie de registros, cada uno correspondiente a un estudiante del curso, con los eventos y su información ordenada cronológicamente. La Figura 7 muestra un ejemplo de un JSON correspondiente a un usuario lo podemos ver en la Figura 7. Se puede observar que el estudiante ha estado navegando por un documento y después ha estado interaccionando con vídeos.


```
{
  "Usuario":"6424889",
  "Eventos":[
    {
      "id_documento":"1.1. El entorno de
      desarrollo de Android",
      "evento":"textbook.pdf.chapter.navigated",
      "tiempo":"2015-02-24T06:59:22.369113+00:00
      "
    },
    {
      "evento":"load_video",
      "tiempo":"2015-02-24T07:00:31.475076+00:00
      ",
      "id_video":"31"
    },
    {
      "evento":"play_video",
      "tiempo":"2015-02-24T07:01:05.638093+00:00
      ",
      "id_video":"31",
      "currentTime":"0"
    },
    {
      "evento":"pause_video",
      "tiempo":"2015-02-24T07:01:28.122204+00:00
      ",
      "id_video":"31",
      "currentTime":"21.232"
    },
```


Figura 7. Eventos preprocesados para un usuario

3.4 Preprocesado para diferentes ediciones

Una de las tareas de este trabajo consiste en poder aplicar el análisis y la predicción de datos a varias ediciones del curso online que estemos analizando. Para ello, es necesario mapear los ids de los problemas y los vídeos, y los tipos de eventos de las distintas ediciones para preprocesar los datos de forma uniforme para todas las ediciones.

Para identificar los ids de los vídeos y los problemas con mayor facilidad, y con la finalidad de llevar a cabo una limpieza de los datos, los ids representados en el log que tienen el siguiente formato para el curso *Jugando con Android – Aprende a programar tu primera App*:

"*i4x://UAMx/Android301x/problem/768d466a8b6b4bcaaffc12e99bfd68fb*"

se sustituyen por enteros del 1 al X en función de la cronología del vídeo o problema. Todo esto se realiza gracias a un mapeado en un fichero JSON que contiene el id del vídeo en el log y el id nuevo. Este fichero de mapeado nos permite hacer coincidir los ids de las diferentes ediciones del curso. Esto es necesario ya que de una edición del curso a otra puede haber cambios en los problemas o vídeos así como en los ids. Por lo tanto, si queremos llevar a cabo un sistema de predicción debemos basarnos en aquellos elementos comunes entre ambas ediciones que serán sobre los que creemos el modelo.

4 Automatización del flujo de datos

Otro de los objetivos principales de este trabajo consiste en la creación de un programa que sea capaz de extraer características de manera automática para cualquier curso online cuya estructura de fichero de entrada sea similar a la del JSON extraído en la fase de preprocesado.

Para llevar a cabo esta tarea se ha realizado un análisis del flujo de datos con la finalidad de establecer un programa que permita, con la mayor sencillez y flexibilidad posible, añadir funcionalidad para extraer nuevas características de los cursos online. Para ello se ha establecido una estructura basada en un diccionario de dataframes cuya clave es la semana del curso y cuyo valor es el dataframe con las características de los estudiantes. Estos dataframes son estructuras de datos similares a las tablas de SQL y que son proporcionadas por la librería *pandas* de Python. Además de los dataframes, se utiliza la librería *numpy* de Python como herramienta para el manejo de datos en arrays.

4.1 Diseño del flujo de datos

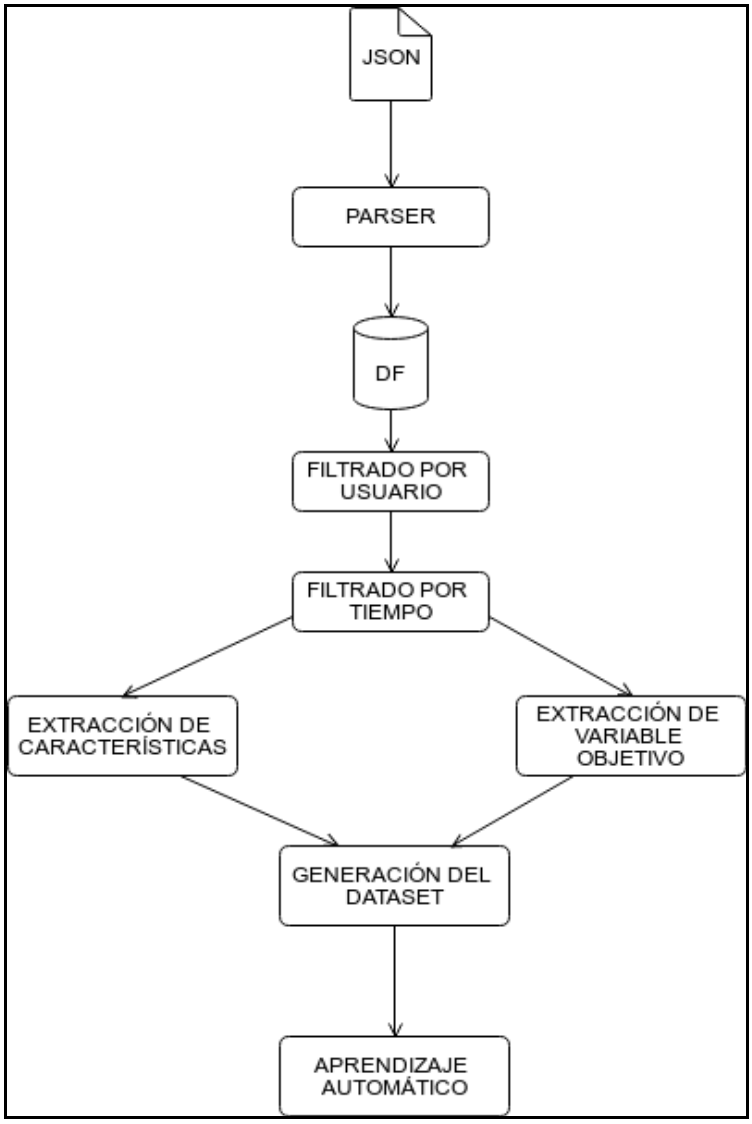

Figura 8. Diseño del flujo de datos

En la Figura 8 se puede observar el diseño del flujo de datos. En primer lugar, disponemos de un fichero con las interacciones de los estudiantes. A partir de ese fichero mediante una función parser, extraemos un dataframe formado por los estudiantes y sus eventos. A continuación, podemos realizar operaciones de filtrado de usuario, por ejemplo, quedarnos con los estudiantes que han hecho más de un número de eventos determinado. Después se realiza el filtrado por tiempo mediante el cual se extraen diferentes dataframes para cada intervalo de tiempo pasado como argumento. Esos dataframes estarán formados por los eventos realizados por cada estudiante en dichos intervalos de tiempo. A partir de los dataframes anteriores se puede llevar a cabo la extracción de características o la extracción de la variable objetivo y finalmente generar el dataset correspondiente. Ese dataset será utilizado por el módulo de aprendizaje automático que esta implementado de manera independiente al curso online analizado.

4.2 Filtrado por eventos y creación de diccionarios

En primer lugar, se filtra por estudiante eliminando aquellos que han realizado menos de un número eventos determinado. Esto debe a que consideramos que los estudiantes que han realizado menos de *x* eventos no están siguiendo el curso sino que simplemente se han inscrito y han navegado por el curso sin mayor interés por continuar en él. Por lo tanto, estos estudiantes nos pueden generar ruido a la hora de llevar a cabo la predicción del abandono o la calificación.

En segundo lugar, se crean dataframes por semanas estableciendo las fechas de cada semana en cada una de las ediciones y se filtra a aquellos estudiantes que hayan realizado mínimo algún evento entre las fechas correspondiente a las semanas del curso.

Todo este proceso de filtrado de estudiantes y creación del diccionario con los dataframes de eventos por semana queda automatizado de manera que solo es necesario pasar las fechas de las semanas de la edición y el número de eventos por el que se quiere filtrar.

4.3 Extracción de características

Una vez se han extraído los eventos por semana en los dataframes correspondientes, se lleva a cabo una extracción de características que nos permitan predecir el abandono y la calificación de los estudiantes.

Este proceso de extracción de las características queda automatizado de manera que no es necesario llevar a cabo ninguna modificación del programa aunque se trate de otra edición del curso. Simplemente se pasa el diccionario de dataframes con los eventos por semana extraído anteriormente, y el programa se encarga de llevar a cabo la extracción de las características.

En el caso de querer añadir una nueva característica, habría que implementar una función adicional pero que puede usar funciones generales como por ejemplo filtrar por evento o contar el número de eventos con la finalidad de conseguir la reutilización de código y la flexibilidad del programa. Para este trabajo se han considerado dos conjuntos de características: generales, que tienen en cuenta número de eventos de cada tipo, y de problemas que solo tienen en cuenta la interacción de los estudiantes con los problemas.

- **Características generales:** Para este análisis, en primer lugar, se extraen las características generales como el número total de eventos, el número de interacciones de vídeo etc, todas ellas se detallarán más adelante en el apartado *5.2*.

Todo esto se realiza para todas las semanas del curso de manera que una vez termina este proceso disponemos de un diccionario de dataframes por cada característica extraída. Estos diccionarios se unen al final para formar un único diccionario de dataframes con las características generales por semana.

- **Características de problemas:** En segundo lugar se extraen los resultados de los ejercicios de los problemas como unas características independientes de las características generales extraídas anteriormente. Este proceso consiste en ver qué problemas ha hecho el estudiante durante la semana actual y a partir de ellos crear un dataframe por semana con los problemas que ha realizado cada usuario, independientemente de si el problema corresponde a esa semana o no. Por ejemplo, un estudiante puede hacer un problema de la semana 1 en la semana 3. En este caso solo se tendría en cuenta para la predicción de la semana 4. Es decir, solo se va a usar la información que está disponible en un momento dado para hacer la predicción de la semana siguiente.

4.4 Definición de abandono y calificación

Para poder modelar una situación real en la que sabemos qué estudiantes abandonan y cual no, hemos considerado dos formas de predecir el abandono en función de los eventos que han realizado los estudiantes. Tal y como hemos mencionado en el estado del arte disponemos del *criterio 1* en el que se considera que un estudiante ha abandonado si no ha realizado ninguna actividad evaluable en las dos semanas siguientes, y el *criterio 2*, en el que consideramos que un estudiante abandona si no realiza ningún evento en la siguiente semana.

Para llevar a cabo la automatización del flujo de datos se han creado dos funciones para cada tipo de criterio que solo requieren del dataframe con los eventos para calcular que estudiantes se considera que abandonan y cuáles no. A partir de aquí tenemos un diccionario de dataframes independiente al de características con los estudiantes y su variable objetivo para cada semana.

Del mismo modo que generamos la etiqueta de abandono, también disponemos de una función para llevar a cabo la extracción de la calificación del examen final de los estudiantes, esta nota es la misma para cada semana y la extraemos de un fichero externo de manera que solo es necesario modificar ese fichero en función de la edición para poder obtener la calificación de los estudiantes.

4.5 Combinación de dataframes

Una vez se han extraído las características con la variable objetivo se deben combinar de manera que para cada semana se disponga de un dataframe con las características generales de esa semana y las de las anteriores semanas. En caso del resultado de los problemas en los que se tiene en cuenta su temporalidad, no es necesario combinarlos entre sí ya que, como hemos mencionado anteriormente para la semana actual ya tenemos en cuenta si el usuario ha realizado problemas de las semanas anteriores.

Tras obtener las características generales, se añade la variable objetivo en cada uno de los dataframes de características por semanas. Todo esto se realiza de manera automática y en una sola ejecución. Después se guardan en un fichero los dataframes correspondientes por semana con el formato csv.

El resultado de este procesado se puede observar en la Figura 9 donde se muestran las características por estudiante y semana y al final se encuentra la variable objetivo del abandono.

Usuario	numCorrectAE	NumEventos_Week1	NumInteraccVideos_Week1	NumProbl	NumProy	NumF	numIntera	Total_wor	self-evaluationPoints_	numAttemps_	numCorrect_Week1	target
1001424	8	308	264	28	0	2	0	0	0	28	21	1
1010202	8	58	39	16	0	0	0	0	0	16	10	0
1023758	5	40	0	36	0	0	0	0	0	40	34	0
103799	9	12	8	2	0	0	0	0	0	2	1	1
1039725	10	191	133	30	0	1	0	0	0	32	23	0
1040408	0	238	224	5	0	0	0	0	0	7	5	0
1041154	9	206	166	29	0	0	0	0	0	31	22	0
1051263	7	159	110	25	0	0	0	0	0	25	21	0
1060716	10	53	31	16	0	0	0	0	0	16	12	0
10750	0	51	40	3	0	1	0	0	0	3	2	0
1077245	7	58	32	20	0	0	0	0	0	23	13	1
1080030	13	73	36	28	0	0	0	0	0	28	26	1
1081715	6	782	696	32	0	0	0	0	0	36	18	0
1089987	11	80	31	24	0	9	0	0	0	24	19	1
1092078	3	55	36	9	0	0	0	0	0	9	6	0
1092094	12	379	288	29	0	2	0	0	0	30	27	1
109670	10	4	3	0	0	0	0	0	0	0	0	0
109823	8	63	25	22	0	5	0	0	0	22	16	1
110500	9	65	32	17	0	0	0	0	0	17	12	0
1109013	0	509	422	12	0	35	0	0	0	12	7	1
1109393	0	132	106	8	0	1	0	0	0	8	7	1
1117682	2	43	27	9	0	0	0	0	0	9	2	1
112138	12	108	63	36	0	0	0	0	0	38	35	1
1130326	10	107	75	20	0	0	0	0	0	20	17	0
113479	10	558	465	43	0	0	0	0	0	45	41	0
1136260	7	75	50	18	0	0	0	0	0	19	10	0
113752	5	136	76	26	0	11	0	0	0	26	15	1
115462	5	93	71	11	0	0	0	0	0	11	11	0
115593	10	72	43	16	0	5	1	0	0	16	14	0

Figura 9. Características generales predicción

5 Aplicación de técnicas de aprendizaje

A continuación vamos a describir los algoritmos de aprendizaje automático, los datos usados para la predicción, el método seguido para predecir el abandono y la calificación de los estudiantes en el examen final, y las métricas que nos sirven para estimar cómo de buenos son nuestros resultados.

Para llevar a cabo esta aplicación de técnicas de aprendizaje automático se utilizan dos ediciones del curso online: *Jugando con Android – Aprende a programar tu primera App*, impartido por profesores de la Universidad Autónoma de Madrid en la plataforma edX.

5.1 Descripción de algoritmos de aprendizaje

A continuación se va a detallar el funcionamiento de los diferentes algoritmos de aprendizaje automático utilizados en este trabajo.

5.1.1 Random Forest

Random forest [29] es un algoritmo de predicción supervisado que combina árboles de decisión mediante voto por mayoría. Esta técnica de conjuntos de clasificadores, como *bagging*, consiste en combinar los resultados de varios modelos con la finalidad de obtener un mejor ajuste que si se utilizase solamente un único modelo.

En este algoritmo se crea un conjunto de árboles de decisión donde se utiliza un subconjunto aleatorio de atributos para cada partición del árbol. Se suele usar la raíz cuadrada del total de atributos como el número atributos aleatorios. Adicionalmente, para generar cada árbol de decisión se utilizará muestreo aleatorio con reemplazamiento de tamaño igual al número de los datos de entrenamiento. Finalmente se realiza la predicción de la clase en cada uno de los árboles generados y una vez se obtienen todas ellas se extrae la clase mayoritaria.

Alguna de las características más relevantes de *Random Forest* son las siguientes:

- Se ejecuta de manera eficiente para grandes cantidades de datos.
- Posee gran precisión respecto a otros algoritmos de predicción actuales.
- Puede manejar miles de variables.
- Da estimaciones de qué variables son importantes en la clasificación.
- Los bosques generados pueden ser guardados para utilizarlos posteriormente con otros datos.

5.1.2 Regresión logística y lineal

Regresión logística es un algoritmo que permite determinar la relación entre las características y una variable dicotómica que es la clase. Para ello este algoritmo se basa en una combinación lineal:

$$Z = b_1 x_1 + b_2 x_2 + \ldots + b_p x_p + b_0$$

En esta fórmula las ***x*** representan las diferentes características, las ***b*** representan los coeficientes del hiperplano, y la *p* hace referencia al índice de una característica concreta. Para saber a qué clase pertenece un patrón de características se calcula la probabilidad de pertenecer a la clase 1 mediante la fórmula de la sigmoidal:

$$P(C1|\ x) = \frac{1}{1+e^z}$$

Durante el entrenamiento del algoritmo estima el valor de los coeficientes del hiperplano, ***b***, mediante un algoritmo iterativo. Una vez estimados estos coeficientes, cada patrón de los datos de test pasará por la combinación lineal anterior y se extraerá el valor Z. Con ese valor de Z vamos a la fórmula de la sigmoidal y sustituimos en el valor del exponente de la *e*. Finalmente si la probabilidad P para esa muestra es mayor de 0,5 se le asignará la clase 1 y si es menos de 0,5 la clase 2.

Regresión lineal es un algoritmo de regresión que sigue el mismo proceso que regresión logística salvo que la salida del modelo no es una probabilidad, simplemente se extrae el valor de la Z que será la predicción numérica.

El método de regresión lineal se utiliza para predecir la calificación, que es un problema de regresión, mientras que regresión logística para la predicción de abandono, que es un problema de clasificación.

5.1.3 XGBoost

XGBoost es un algoritmo basado *gradient boosting* que consiste en combinar las contribuciones de múltiples árboles de decisión sencillos mediante voto [30].

XGBoost se encarga de crear varios árboles de decisión cada uno con diferentes características aleatorias y promediando el resultado de cada uno de ellos. La diferencia entre *Random Forest* y *XGBoost* es que en *Random Forest* los árboles que se crean son independientes entre sí mientras que en *XGBoost* el entrenamiento de los árboles se hace de manera secuencial ya que se tiene en cuenta el error de cada árbol con la finalidad de mejorarlo en el siguiente árbol.

5.2 Descripción de los atributos

Para este sistema es necesario disponer de unos ficheros csv con el formato explicado en el apartado anterior (Figura 9). Este fichero csv posee los datos de las características acumuladas para cada usuario según las semanas, dispondremos de tres tipos de ficheros csv.

El primer tipo fichero (que es igual que la Figura 9) contiene características generales de interacciones de los estudiantes con el curso:

- Número de eventos total: esta característica corresponde al número de eventos realizados por un estudiante por semana.

- Número de interacciones de vídeo: esta característica consiste en el número de veces que el estudiante interacciona con un evento de vídeo de los siguientes: *play_video, pause_video, seek_video, stop_video*.

- Número de interacciones de problemas: esta característica cuenta el número de veces que un estudiante interacciona con los problemas por semana.

- Número de problemas correctos: esta característica cuenta el número de veces que un estudiante hace correctamente los ejercicios de los problemas por semana.

- Número de intentos en problemas: esta característica cuenta el número de intentos que ha realizado el estudiante por semana.

- Número de aciertos en la autoevaluación sobre java: este curso online que estamos analizando dispone de una autoevaluación de java durante las primeras semanas de curso que consta de 14 problemas. Esta característica cuenta cuántos de estos problemas ha resuelto correctamente el estudiante.

- Puntos proyectos: esta característica suma los puntos de la autoevaluación del proyecto por semana.

- Número de interacciones con proyecto: esta característica cuenta el número de veces que el estudiante hace un proyecto por semana.

- Número de palabras foro: esta característica cuenta el número de palabras que ha escrito el estudiante en el foro contemplando los eventos de creación de hilo, respuestas a hilos y comentarios a respuestas.

- Número de interacciones con documentos: esta característica cuenta el número de veces que el estudiante interacciona con un documento por semana.

- Número de interacciones con foro: esta característica cuenta el número de veces que el estudiante interacciona con el foro del curso por semana.

En el segundo tipo de fichero guardaremos las características de los problemas en las que vamos a tener en cuenta la temporalidad de los mismos, es decir a que semana pertenece el problema y su orden a la hora de extraer sus características. Para ello contamos con que el id del problema viene relacionado con su temporalidad tal y como hemos explicado anteriormente. Por lo tanto, se van a extraer como características el número de intentos que hace el estudiante sobre un problema y si los ejercicios que lo componen son correctos, incorrectos o no se han realizado. Esto quedará representado mediante el siguiente mapeado de valores: 0 para el ejercicio fallado, 1 para el ejercicio correcto y -1 para el ejercicio no realizado. Un ejemplo del formato de ese fichero es el de la Figura 10.

Usuario	Intentos_1_1	Problema_1_1	Intentos_2_1	Problema_2_1	Intentos_3_1	Problema_3_1	Intentos_4_1	Problema_4_1	Intentos_5_1	Problema_5_1	Intentos_6_1	Problema_6_1	Intentos_7
1001424	1	1	1	0	1	0	1	1	1	1	1	0	
1010202	1	1	1	1	1	0	1	1	1	1	1	0	
1023758	1	1	1	1	1	1	1	1	1	1	1	0	
103799	1	1	1	0	0	-1	0	-1	0	-1	0	-1	
1039725	1	1	1	1	1	0	1	1	1	1	1	1	
1040408	0	-1	0	-1	0	-1	0	-1	0	-1	0	-1	
1041154	1	1	1	1	1	0	1	1	1	1	1	0	
1051263	1	1	1	1	1	1	1	1	1	1	1	0	
1060716	1	1	1	0	1	0	1	1	1	1	1	0	
10750	1	0	0	-1	0	-1	0	-1	0	-1	0	-1	
1077245	1	0	1	1	1	1	1	1	1	1	1	0	
1080030	1	1	1	1	1	1	1	1	1	1	1	1	
1081715	1	0	1	1	1	1	1	1	1	1	1	0	
1089987	1	1	1	1	1	1	1	1	1	1	1	0	
1092078	1	1	1	0	1	0	1	1	1	1	1	0	
1092094	1	1	1	1	1	1	1	1	1	1	1	1	
109670	0	-1	0	-1	0	-1	0	-1	0	-1	0	-1	
109823	1	0	1	0	1	1	1	1	1	1	1	0	
110500	1	0	1	1	1	1	1	1	1	1	1	0	
1109013	0	-1	0	-1	0	-1	0	-1	0	-1	0	-1	
1109393	0	-1	0	-1	0	-1	0	-1	0	-1	0	-1	
1117682	1	0	1	0	1	1	1	0	1	1	1	0	
112138	1	1	1	1	1	1	1	1	1	1	1	1	

Figura 10. Formato csv problemas predicción

Como podemos observar en la figura 10 para cada ejercicio tenemos el número de intentos que ha realizado el estudiante y si lo ha resuelto correctamente o no. En la última columna del csv se encontraría la variable objetivo tal y como hemos mencionado para el fichero de la Figura 9.

El último tipo de fichero consiste en la combinación de ambos grupos de características: las generales y las de los resultados de los ejercicios.

En el flujo de los datos se filtran aquellos estudiantes que han realizado menos de 50 eventos. Una vez realizado el filtrado en ambas ediciones del curso comprobamos que el número de estudiantes se reduce considerablemente, en la primera edición se reduce de *7170* estudiantes a *3138* estudiantes y en la segunda edición de *6818* estudiantes a *2647* estudiantes.

Para cada una de las ediciones los números de estudiantes por semanas quedan representados en las Tablas 1 y 2.

	ESTUDIANTES
Semana 1	*2162*
Semana 2	*1627*
Semana 3	*1691*
Semana 4	*1591*
Semana 5	*1142*
Semana 6	*871*

Tabla 1. Número usuarios por semana edición 1

	ESTUDIANTES
Semana 1	*1723*
Semana 2	*1590*
Semana 3	*1444*
Semana 4	*1229*
Semana 5	*994*
Semana 6	*817*

Tabla 2. Número usuarios por semana edición 2

Las Tablas 3 a 6 muestran para cada semana de cada edición los usuarios que empiezan y cuantos consideramos que van a abandonar según los dos tipos de criterio: el estudiante no ha realizado ninguna actividad evaluable en las dos semanas siguientes (criterio 1), el estudiante no realiza ningún evento en la siguiente semana (criterio 2).

	Número Estudiantes	***Abandonan***	***Continúan***
Semana 1	*2162*	*811*	*1351*
Semana 2	*1627*	*544*	*1083*
Semana 3	*1691*	*715*	*976*
Semana 4	*1591*	*916*	*675*

Tabla 3. Evolución semanal estudiantes, para criterio 1 edición 1

	Número Estudiantes	***Abandonan***	***Continúan***
Semana 1	*2162*	*856*	*1306*
Semana 2	*1627*	*494*	*1133*
Semana 3	*1691*	*546*	*1145*
Semana 4	*1591*	*685*	*906*
Semana 5	*1141*	*499*	*643*

Tabla 4. Evolución semanal estudiantes, para criterio 2 edición 1

	Número Estudiantes	***Abandonan***	***Continúan***
Semana 1	*1723*	*527*	*1196*
Semana 2	*1590*	*623*	*966*
Semana 3	*1444*	*627*	*817*
Semana 4	*1229*	*636*	*593*

Tabla 5. Evolución semanal estudiantes, para criterio 1 edición 2

	Número Estudiantes	***Abandonan***	***Continúan***
Semana 1	*1723*	*520*	*1203*
Semana 2	*1590*	*563*	*1026*
Semana 3	*1444*	*540*	*904*
Semana 4	*1229*	*468*	*761*
Semana 5	*994*	*395*	*599*

Tabla 6. Evolución semanal estudiantes, para criterio 2 edición 2

Para predecir la calificación de los estudiantes en el examen final vamos a utilizar la misma combinación de características cambiando únicamente la variable objetivo. Para la primera edición disponemos de la nota del examen de 495 estudiantes, y para la segunda edición de 349. Todos estos estudiantes no tienen por qué aparecer en la primera semana sino que algunos van añadiéndose a lo largo de las semanas siguientes y por lo tanto no podemos empezar a predecir con todos ellos. La evolución semanal de los estudiantes para predecir la nota del examen final se muestra en las tablas 7 y 8.

	Número estudiantes
Semana 1	*382*
Semana 2	*372*
Semana 3	*417*
Semana 4	*465*
Semana 5	*459*
Semana 6	396

Tabla 7. Evolución semanal de los estudiantes para predicción de nota en la edición 1

	Número estudiantes
Semana 1	*270*
Semana 2	*286*
Semana 3	*306*
Semana 4	*311*
Semana 5	*318*
Semana 6	316

Tabla 8. Evolución semanal de los estudiantes para predicción de nota en la edición 2

Se puede observar que en ocasiones el número de estudiantes entre una semana y otra disminuye, esto es debido a que habrá estudiantes que aparezcan en unas semanas y en otras estén inactivos, es decir, que no realizarán ningún evento y por lo tanto no podemos predecirlos durante esa semana. En una situación real consideraríamos que esos estudiantes ya no están en el curso o por lo menos durante esa semana aunque luego realicen el examen final.

Con todos los datos mostrados anteriormente se lleva a cabo la tarea de predicción de abandono y de la calificación cuyo procesado y resultado se muestra en las secciones siguientes.

5.3 Experimentos

A continuación vamos a describir el método de aprendizaje automático que se ha aplicado para llevar a cabo la predicción del abandono y la calificación de los estudiantes. El método consiste en una doble validación cruzada de 10 particiones para la primera edición del curso. Con los mejores resultados obtenidos se realiza la predicción para la segunda edición del curso. Para llevar a cabo esta fase de aprendizaje automático nos basaremos en la librería *sklearn* de Python que contiene gran cantidad de funciones para el aprendizaje automático siendo la de uso más extendido dentro de este lenguaje de programación.

5.3.1 Método

En primer lugar se realiza una validación cruzada que genera 10 particiones. Cada una de ellas consta de entrenamiento y de test. Estas particiones las llamaremos particiones principales. Sobre esta validación cruzada se aplica la función *GridSearchCV* en la parte de entrenamiento de las particiones anteriores. Esta función se encarga de crear otras 10 particiones, que llamaremos particiones secundarias, y a partir de ellas obtiene los mejores parámetros para el clasificador pasado como argumento. En nuestro caso se van a utilizar tres algoritmos para el abandono y otros tres para la predicción de la nota que han sido descritos en el apartado 1 de este capítulo. Los parámetros que va a probar la función *GridSearchCV* son los siguientes en función del algoritmo:

- *Clasificador Regresión logística:* el parámetro *C* representa una penalización para reducir el sobreajuste. Se prueba con los siguientes valores*: [0.1, 1.0, 10]*. Otro de los parámetros a estimar es el *max_iter* que representa el número máximo de iteraciones del algoritmo. Se prueba con los valores *[100, 200]*.

- *Clasificador Random Forest:* el parámetro *n_estimators* representa el número de árboles que poseerá el bosque, en nuestro caso le asignamos el valor de *500*. El parámetro *max_features* hace referencia a la función que se va a aplicar al número de características para ver cuantas se considera tras la división. Se probará con las funciones *[sqrt, log2]*. Por último el argumento *criterion* toma el valor *gini* que hace referencia a la impureza.

- *Clasificador XGBoost:* el parámetro *n_stimators* hace referencia al número de árboles. Se prueba con los valores *[100,200,300]*. Otro parámetro que se estima es *max_depth* que contendrá los valores *[3, 7]* ya que al probar con varios estimadores el tiempo de ejecución aumenta y queremos que los algoritmos no tarden excesivamente para poder realizar el mayor número de pruebas.

- *Regresión lineal:* para este algoritmo regresor utilizaremos el parámetro *normalize* con los valores *[True, False]*. Este parámetro nos sirve para probar si funciona mejor el regresor con los datos normalizados o sin normalizar.

- *Regresor Random Forest:* para este algoritmo utilizaremos los mismos parámetros que en el anterior de clasificación excepto el parámetro *criterion* que toma el valor de *mae*, que hace referencia al error absoluto medio, ya que para medir la calidad de la división necesitamos una métrica de regresión.

- *Regresor XGBoost:* para este algoritmo se van a probar los mismos parámetros que para el clasificador XGBoost.

Otro de los parámetros necesarios por la función *GridSearchCV* es la métrica. Para llevar a cabo la predicción del abandono se ha empleado la métrica *acierto* que consiste en contar el número de veces que son iguales la etiqueta del valor real y el predicho y dividirlo entre el total de datos de test.

$$\frac{\sum_{test}[real = predicha]}{numero\ datos\ test}$$

Para la predicción de la nota la métrica *error absoluto o mae* que mide la diferencia en valor absoluto entre la predicción y el valor real.

$$\frac{\sum_{test}|real - predicha|}{Numero\ datos\ test}$$

Una vez se han extraído los mejores parámetros en las particiones secundarias se aplica el algoritmo con dichos parámetros a cada partición principal obteniéndose así 10 resultados para cada partición. De esos resultados se extraen el *acierto* medio, la desviación típica y nos quedamos con los mejores algoritmos para poder llevar a cabo la predicción de la siguiente edición sin necesidad de tener que realizar el tuneado de parámetros.

Para la predicción de la segunda edición ya no es necesario realizar particiones sino que se entrena con la primera edición y los parámetros obtenidos anteriormente y se predice la segunda edición como conjunto de datos de test.

5.3.2 Predicción del abandono

Hasta ahora hemos explicado el método para llevar a cabo tanto la predicción de la calificación como del abandono, que en ambos casos es el mismo. En el caso del abandono además del *acierto* hemos medido la bondad de nuestro modelo mediante las curvas ROC y su métrica asociada, *el área bajo la curva* (AUC, por sus siglas en inglés).

Las curvas ROC representan gráficamente la calidad de los algoritmos asociando la tasa de falsos positivos frente a los verdaderos positivos. A estas curvas se les asocia la métrica del *área bajo la curva* cuyo valor indica la calidad del algoritmo de predicción.

Para poder llevar a cabo la representación de las curvas se ha recogido el número de falsos positivos y verdaderos positivos de cada uno de los algoritmos para cada una de las semanas y para cada una de las combinaciones de características descritas anteriormente en función del criterio de abandono.

La finalidad de la predicción del abandono ces poder avisar a aquellos estudiantes que van a abandonar el curso y de ayudarles a reengancharse en él, proporcionándoles ayuda extra si fuese necesario.

5.3.3 Predicción de la calificación

Para la predicción de la calificación el método empleado utilizamos la métrica del *error absoluto* con el fin de poder establecer la diferencia entre la calificación predicha y la real. Por otro lado, para poder observar mejor los resultados se han realizado gráficas de evoluciones semanales con la calificación predicha y la real de los usuarios. A través de ellas se observa cómo de bueno es nuestro algoritmo de regresión. Estas gráficas se han realizado por semana para cada algoritmo y para cada una de las combinaciones de características.

5.4 Resultados

A continuación se van a mostrar los diferentes resultados para los experimentos descritos anteriormente. Estos experimentos hacen referencia a la predicción de la calificación y del abandono de los estudiantes por semana, criterio, edición y algoritmo.

5.4.1 Resultados abandono criterio 1 edición 1

A continuación vamos a incluir las tablas y gráficas con los resultados obtenidos para la predicción de abandono para el criterio de abandono 1 (los usuarios no realizan ninguna actividad evaluable en dos semanas) y la edición 1 del curso.

La diferencia entre las Tablas 9, 10 y 11, consiste en las características utilizadas para llevar a cabo la predicción del abandono en el criterio 1 y para la edición 1.

Problemas						
	LogReg	DesvTípica	RFC	DesvTípica	XGB	DesvTípica
Semana 1	0,629	0,016	0,618	0,017	0,616	0,028
Semana 2	0,652	0,016	0,645	0,019	0,637	0,017
Semana 3	0,684	0,053	0,668	0,052	0,67	0,047
Semana 4	0,835	0,031	0,844	0,021	0,844	0,023

Tabla 9. Resultados utilizando las características de problemas, edición 1 criterio 1

En la tabla 9 cada columna corresponde al algoritmo utilizado: *LogReg* significa regresión logística, *RFC*, random forest y *XGB*, xgboost. En cada una de ellas se indica el acierto del algoritmo por semana junto con la desviación típica ya que ese acierto representa la media de 10 particiones tal y como se ha explicado anteriormente. Además, se colorea la celda de un color verde más claro para aquel algoritmo que ofrece mejor acierto por semana ya que en una situación real utilizaríamos únicamente ese algoritmo para predecir el abandono en la siguiente edición.

Puede observarse que, a partir de la última semana, el acierto en la predicción de abandono aumenta considerablemente. Esto puede deberse a que, al tratarse de las últimas semanas del curso, la mayor parte de los estudiantes que realizan problemas en esta edición los seguirán haciendo en las próximas dos semanas. Esto no ocurrirá así para los estudiantes que abandonan. Predecimos bien ambos tipos de estudiantes para esa semana como se puede observar en las curvas *ROC* de la figura 11.

En estas curvas se puede observar cómo van evolucionando nuestros modelos de predicción a lo largo de las semanas. En cada gráfica en la parte inferior derecha se encuentra *el área bajo la curva* de cada clasificador. Esta área nos indica la bondad de nuestro modelo de 0 a 1 tal y como hemos explicado anteriormente. Como se ha mencionado anteriormente, en la Figura se observa que en la cuarta semana del curso la predicción de abandono es muy exacta.

Las tablas 10 y 11 muestran los resultados para las características generales y la combinación entre características generales y problemas.

Características generales						
	LogReg	DesvTípica	RFC	DesvTípica	XGB	DesvTípica
Semana 1	0,636	0,028	0,616	0,033	0,641	0,35
Semana 2	0,683	0,034	0,674	0,044	0,667	0,35
Semana 3	0,681	0,036	0,698	0,031	0,685	0,04
Semana 4	0,859	0,031	0,864	0,026	0,851	0,027

Tabla 10. Resultados utilizando las características generales, edición 1 criterio 1

Problemas + Características generales						
	LogReg	DesvTípica	RFC	DesvTípica	XGB	DesvTípica
Semana 1	0,632	0,021	0,634	0,025	0,653	0,022
Semana 2	0,657	0,026	0,674	0,042	0,669	0,038
Semana 3	0,678	0,054	0,69	0,044	0,688	0,041
Semana 4	0,852	0,028	0,871	0,02	0,864	0,02

Tabla 11. Resultados utilizando las características generales y problemas, edición 1 criterio 1

Como se puede observar tras analizar las tablas 9, 10, 11 el mayor acierto se obtiene, por lo general en las 10 y 11 que son muy parecidas. Para decidir cuál de las dos es mejor recurrimos a las curvas ROC de las figuras 12 y 13.

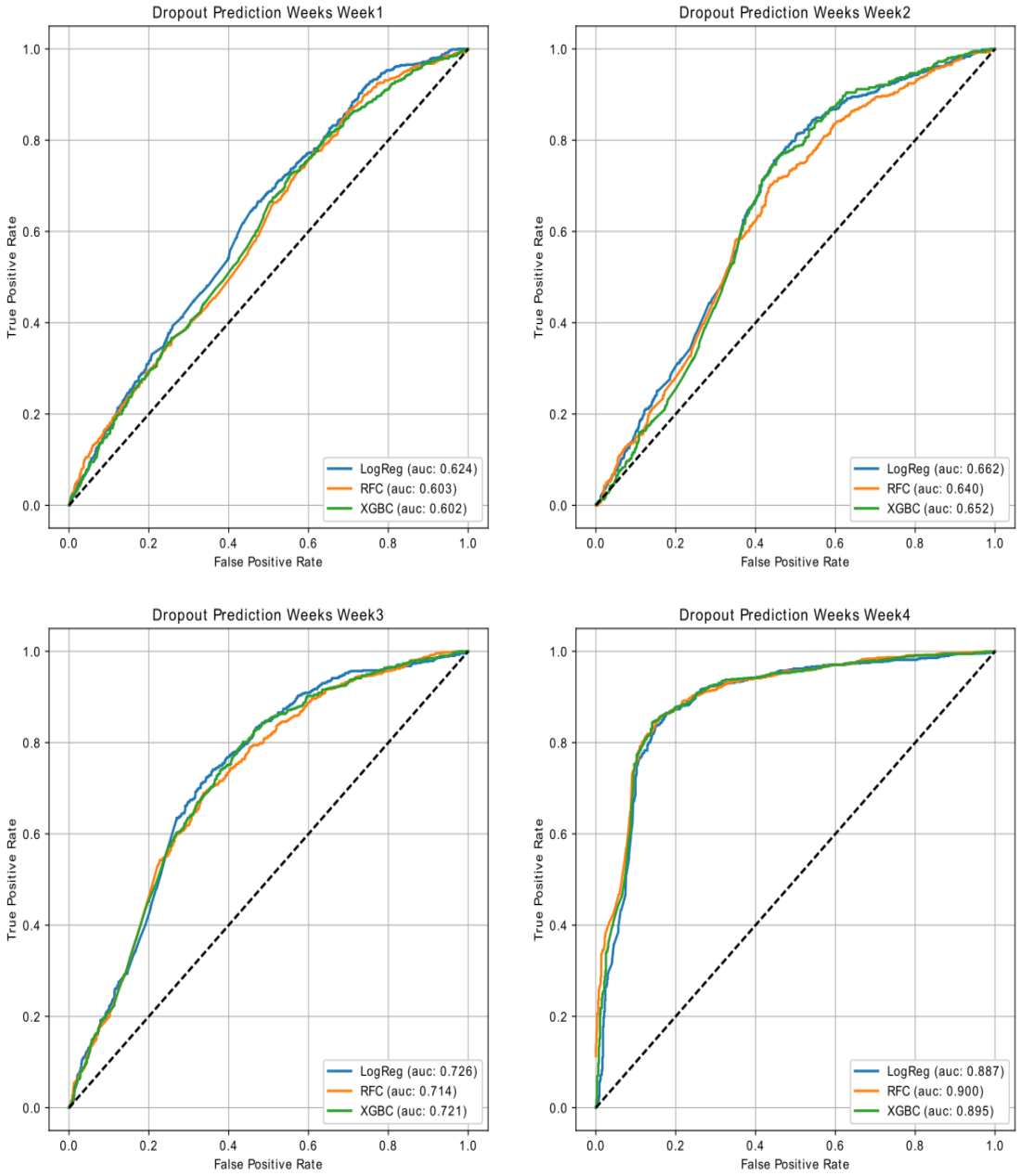

Figura 11. Curvas ROC utilizando las características de problemas, edición 1 criterio 1

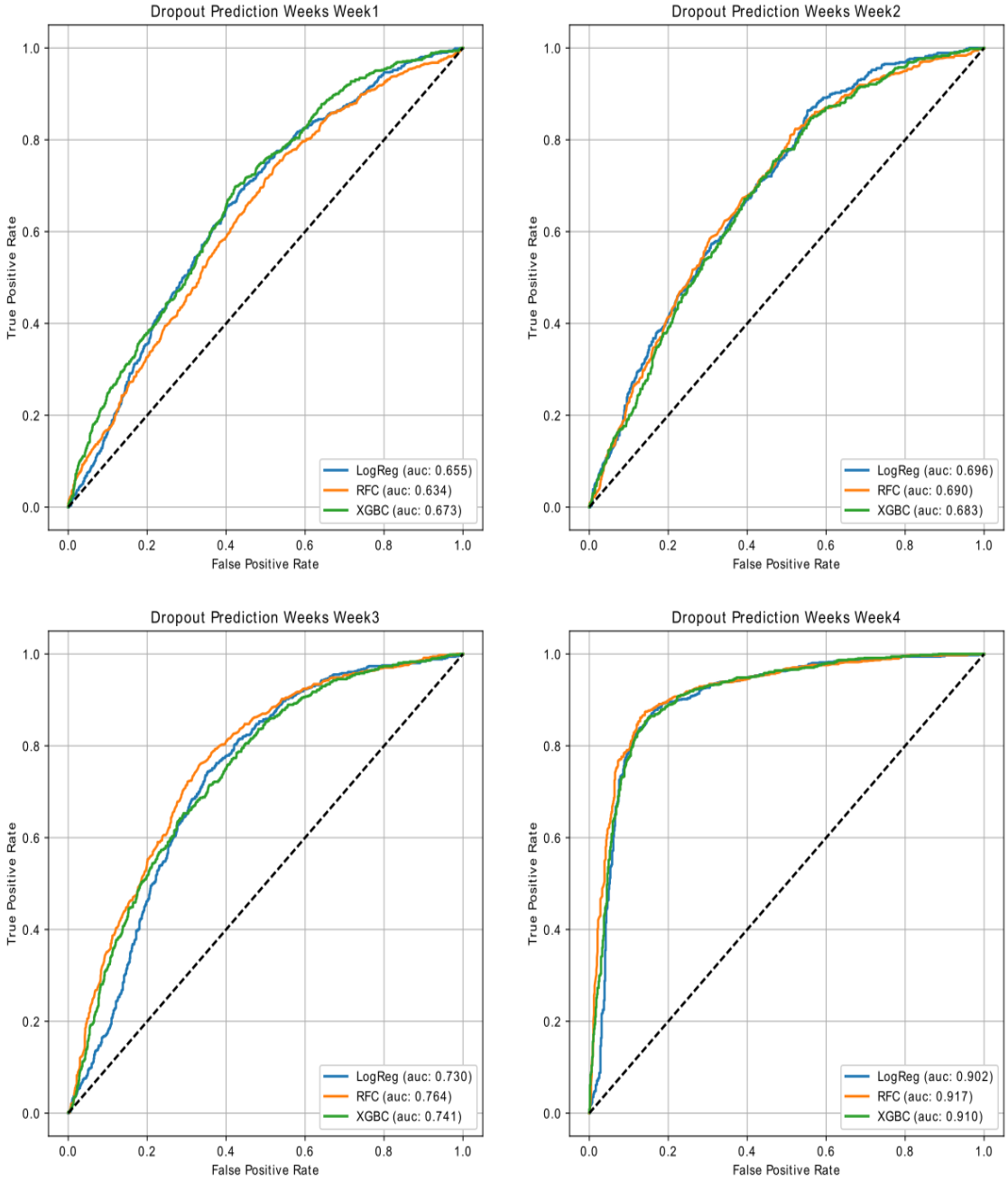

Figura 12. Curvas ROC utilizando las características generales, edición 1 criterio 1

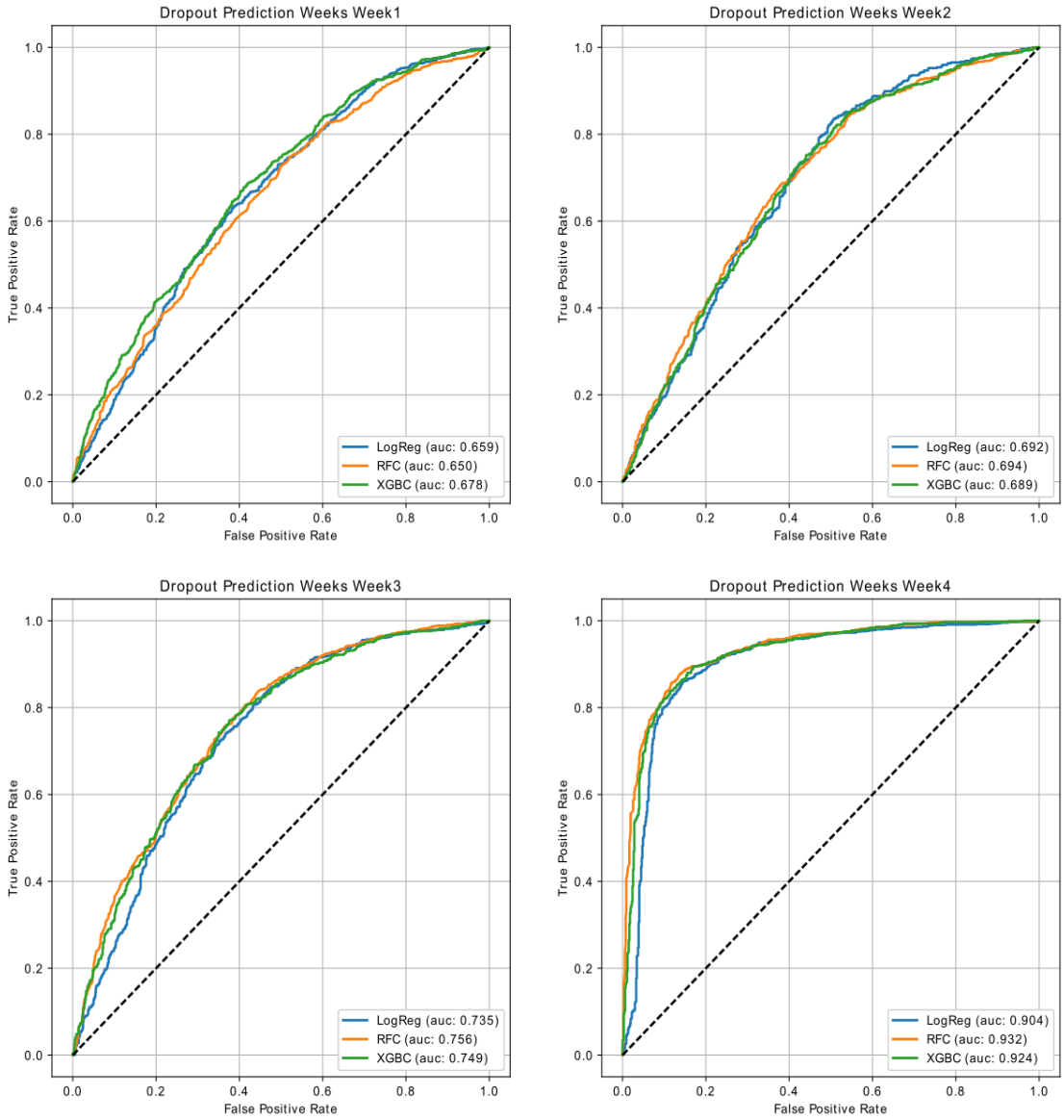

Figura 13. Curvas ROC utilizando las características generales y problemas, edición 1 criterio 1

Analizando el área bajo la curva de las figuras 12 y 13 podemos concluir que la combinación de las características generales y de problemas mejora un poco la predicción del abandono según el criterio 1 aunque las diferencias son mínimas.

5.4.1 Resultados abandono criterio 2 edición 1

A continuación vamos a extraer los resultados para el criterio 2 (se considera que los usuarios abandonan si no han realizado ninguna interacción en la semana siguiente). Por lo tanto, aquí disponemos de una semana más a la hora de aplicar el método de predicción.

Las tablas y gráficas que vamos a mostrar a continuación poseen el mismo formato que las anteriores.

Problemas						
	LogReg	DesvTípica	RFC	DesvTípica	XGB	DesvTípica
Semana 1	0,6	0,024	0,606	0,023	0,6	0,027
Semana 2	0,683	0,011	0,662	0,017	0,668	0,023
Semana 3	0,679	0,022	0,671	0,033	0,674	0,019
Semana 4	0,718	0,049	0,719	0,052	0,726	0,05
Semana 5	0,678	0,048	0,653	0,048	0,664	0,045

Tabla 12. Resultados utilizando las características de problemas, edición 1 criterio 2

Características generales						
	LogReg	DesvTípica	RFC	DesvTípica	XGB	DesvTípica
Semana 1	0,589	0,028	0,597	0,02	0,632	0,031
Semana 2	0,698	0,014	0,686	0,036	0,698	0,023
Semana 3	0,649	0,023	0,699	0,035	0,688	0,043
Semana 4	0,735	0,054	0,735	0,045	0,725	0,045
Semana 5	0,685	0,046	0,691	0,047	0,669	0,039

Tabla 13. Resultados utilizando las características generales, edición 1 criterio 2

Problemas + Características generales						
	LogReg	DesvTípica	RFC	DesvTípica	XGB	DesvTípica
Semana 1	0,604	0,027	0,618	0,024	0,628	0,024
Semana 2	0,682	0,027	0,683	0,032	0,684	0,038
Semana 3	0,671	0,028	0,694	0,049	0,693	0,047
Semana 4	0,735	0,055	0,741	0,043	0,744	0,05
Semana 5	0,673	0,038	0,679	0,052	0,656	0,054

Tabla 14. Resultados utilizando las características generales y problemas, edición 1 criterio 2

Si se observan las tablas 12,13 y 14 se puede ver que, en un principio, las características generales son las que ofrecen mejores resultados. Sin embargo, para poder valorar mejor este resultado vamos a recurrir a la visualización de las curvas ROC junto con el área bajo la curva.

En las figuras 14, 15 y 16 se observa que los problemas por si solos no nos proporcionan información útil. Sin embargo, al incluirlos junto con las características generales, los resultados mejoran considerablemente. De todos modos esta combinación de características generales y problemas obtiene resultados muy similares o incluso peores que si se usan solo las características generales. Por lo tanto, añadir más atributos con la finalidad de mejorar un poco o nada no es una buena decisión. Una opción para llevar a

cabo la predicción de otra edición con este criterio puede ser utilizar únicamente las características generales.

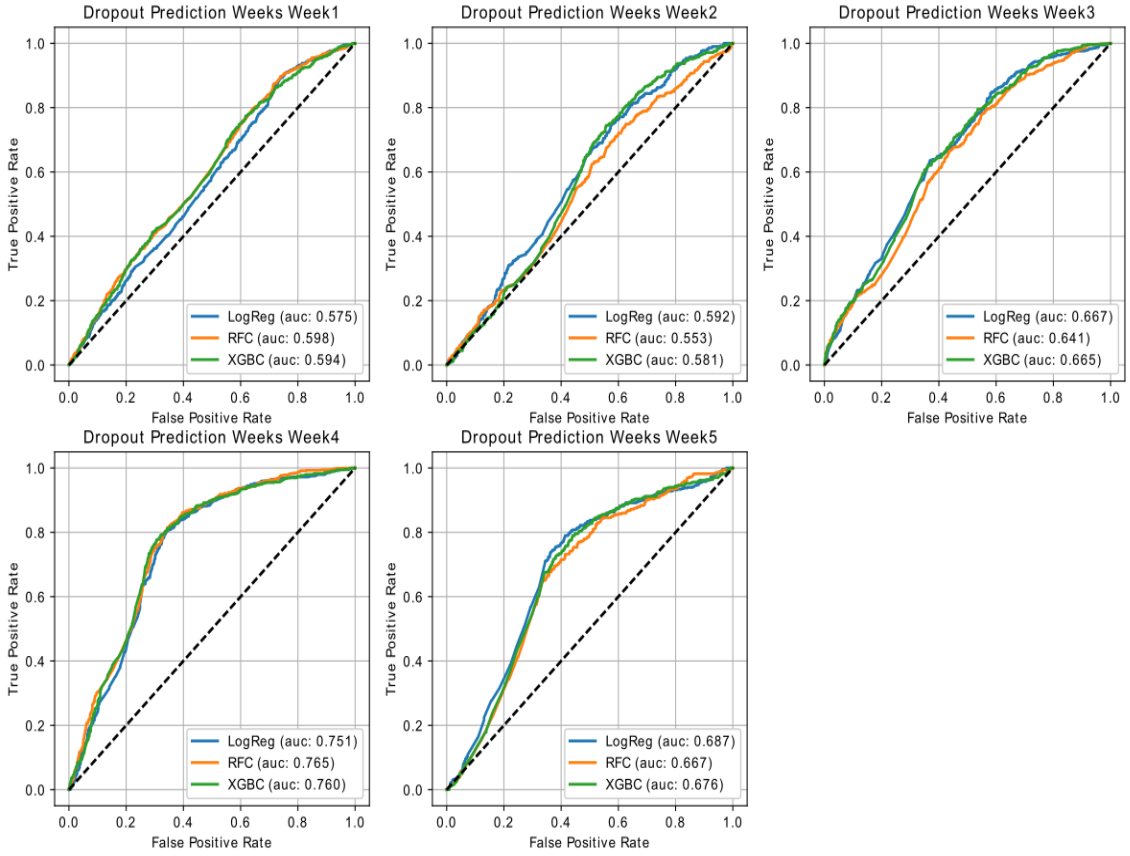

Figura 14. Curvas ROC utilizando características de problemas, edición 1 criterio 2

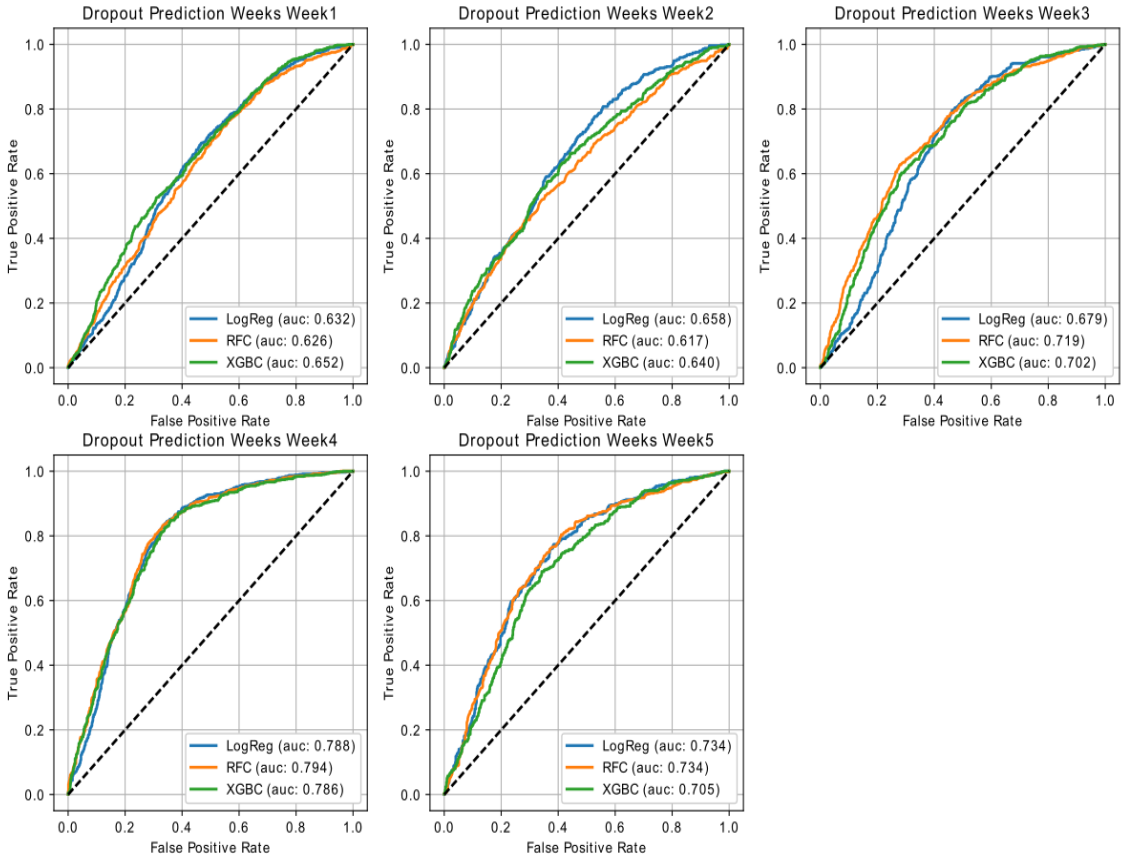

Figura 15. Curvas ROC utilizando las características generales, edición 1 criterio 2

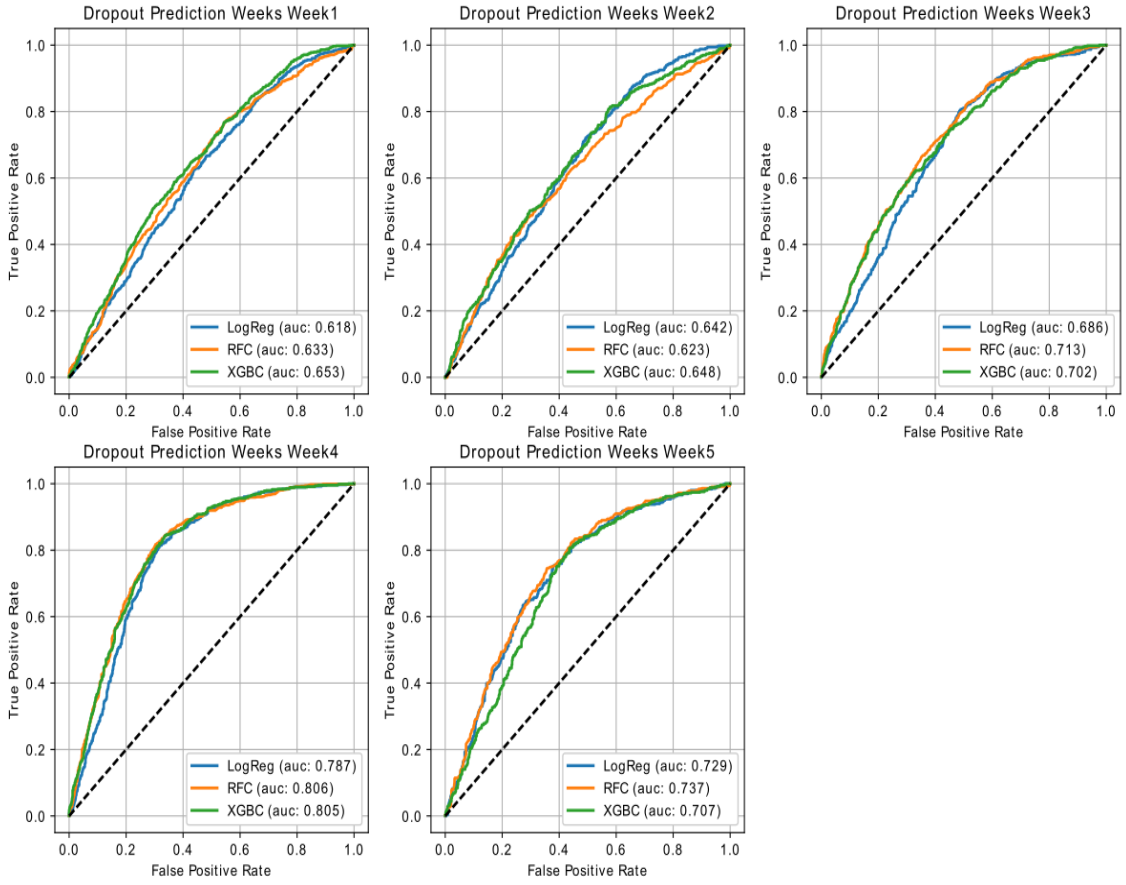

Figura 16. Curvas ROC utilizando las características generales y de problemas, edición 1 criterio 2

5.4.2 Resultados abandono criterio 1 edición 2 a partir de los datos de la edición 1

A continuación mostraremos los resultados del *acierto* en la predicción del abandono en la edición 2 utilizando modelos que han sido entrenados con datos de la edición 1. Es decir, esto sería la configuración más parecida a un entorno real, en la que no puedes entrenar los modelos con los datos del curso en curso. Para esto hemos extraído los mejores modelos con los mejores parámetros utilizando los datos de la edición 1, y hemos utilizado como test la nueva edición.

Como se puede observar en las tablas 15, 16 y 17 el acierto en la segunda edición disminuye con respecto al que teníamos en la cuarta semana de la primera edición. Esto es debido a que al tratarse de otra nueva edición el conjunto de estudiantes no tiene por qué interaccionar de la misma forma que los de la primera edición y, por lo tanto, es esperable que el *acierto* pueda disminuir. Es importante hacer notar que cuando se predice con datos

de la misma edición los estudiantes son los mismos y se espera que sus comportamientos sean contantes a lo largo del curso.

Problemas			
	LogReg	RFC	XGB
Semana 1	0,656	0,625	0,622
Semana 2	0,624	0,619	0,624
Semana 3	0,682	0,695	0,698
Semana 4	0,653	0,661	0,659

Tabla 15. Resultados utilizando características de problemas, edición 2 criterio 1

Características generales			
	LogReg	RFC	XGB
Semana 1	0,679	0,665	0,695
Semana 2	0,651	0,679	0,678
Semana 3	0,719	0,723	0,711
Semana 4	0,706	0,713	0,718

Tabla 16. Resultados utilizando las características generales, edición 2 criterio 1

Problemas + Características generales			
	LogReg	RFC	XGB
Semana 1	0,653	0,677	0,68
Semana 2	0,652	0,683	0,671
Semana 3	0,693	0,732	0,716
Semana 4	0,681	0,676	0,698

Tabla 17. Resultados utilizando las características generales y de problemas, edición 2 criterio 1

5.4.1 Resultados abandono criterio 2 edición 2 a partir de datos de la edición 1

Al igual que hemos hecho anteriormente y siguiendo la misma mecánica vamos a mostrar los resultados para el criterio 2.

Problemas			
	LogReg	RFC	XGB
Semana 1	0,632	0,598	0,637
Semana 2	0,64	0,627	0,636
Semana 3	0,644	0,641	0,653
Semana 4	0,592	0,584	0,597
Semana 5	0,586	0,597	0,593

Tabla 18. Resultados utilizando características de problemas, edición 2 criterio 2

Características generales			
	LogReg	RFC	XGB
Semana 1	0,667	0,613	0,631
Semana 2	0,646	0,656	0,631
Semana 3	0,651	0,674	0,679
Semana 4	0,659	0,65	0,643
Semana 5	0,667	0,683	0,667

Tabla 19. Resultados utilizando las características generales, edición 2 criterio 2

Problemas + Características generales			
	LogReg	RFC	XGB
Semana 1	0,616	0,691	0,62
Semana 2	0,648	0,652	0,641
Semana 3	0,659	0,677	0,675
Semana 4	0,646	0,623	0,631
Semana 5	0,676	0,691	0,665

Tabla 20. Resultados utilizando las características generales, edición 2 criterio 2

En las tablas 18, 19 y 20 se pueden extraer las mismas conclusiones que para el criterio 1. Es decir, que al tratarse de una nueva edición el *acierto* disminuye al haber entrenado los modelos con datos de otra edición.

5.4.2 Predicción calificación edición 1

A continuación en las tablas 21, 22 y 23 vamos a analizar los resultados relativos a la predicción de la calificación para la edición 1. El método que se ha seguido es similar al de la predicción del abandono pero usando regresores en lugar de clasificadores. La métrica utilizada es el error absoluto medio tal y como hemos mencionado anteriormente.

Problemas						
	LinReg	DesvTípica	RFR	DesvTípica	XGB	DesvTípica
Semana 1	0,109	0,024	0,098	0,026	0,106	0,025
Semana 2	0,139	0,023	0,094	0,023	0,098	0,021
Semana 3	0,161	0,015	0,096	0,022	0,098	0,019
Semana 4	0,197	0,041	0,091	0,025	0,092	0,018
Semana 5	0,11	0,021	0,094	0,025	0,099	0,024
Semana 6	0,112	0,03	0,091	0,025	0,088	0,019

Tabla 21. Resultados calificación utilizando características de problemas, edición 1

Características generales						
	LingReg	DesvTípica	RFR	DesvTípica	XGB	DesvTípica
Semana 1	0,092	0,023	0,101	0,025	0,106	0,024
Semana 2	0,086	0,023	0,091	0,024	0,094	0,023
Semana 3	0,089	0,021	0,094	0,024	0,098	0,023
Semana 4	0,091	0,017	0,097	0,022	0,099	0,02
Semana 5	0,089	0,018	0,096	0,021	0,098	0,02
Semana 6	0,083	0,02	0,088	0,022	0,089	0,019

Tabla 22. Resultados calificación utilizando las características generales, edición 1

Problemas + Características generales						
	LinReg	DesvTípica	RFR	DesvTípica	XGB	DesvTípica
Semana 1	0,108	0,027	0,101	0,025	0,106	0,025
Semana 2	0,144	0,017	0,089	0,024	0,093	0,022
Semana 3	0,15	0,017	0,093	0,023	0,097	0,024
Semana 4	0,232	0,031	0,089	0,022	0,09	0,02
Semana 5	0,112	0,015	0,094	0,022	0,099	0,019
Semana 6	0,113	0,024	0,087	0,022	0,089	0,02

Tabla 23. Resultados calificación utilizando características generales y problemas, edición 1

Las notas de los estudiantes van de 0 a 1 y la métrica es el *error absoluto* por lo que la nota predicha oscila en 1 punto con la nota real. Estos resultados aparecen en la Figura 10 donde el eje X representa la nota real y el eje Y la nota predicha para cada algoritmo por semana.

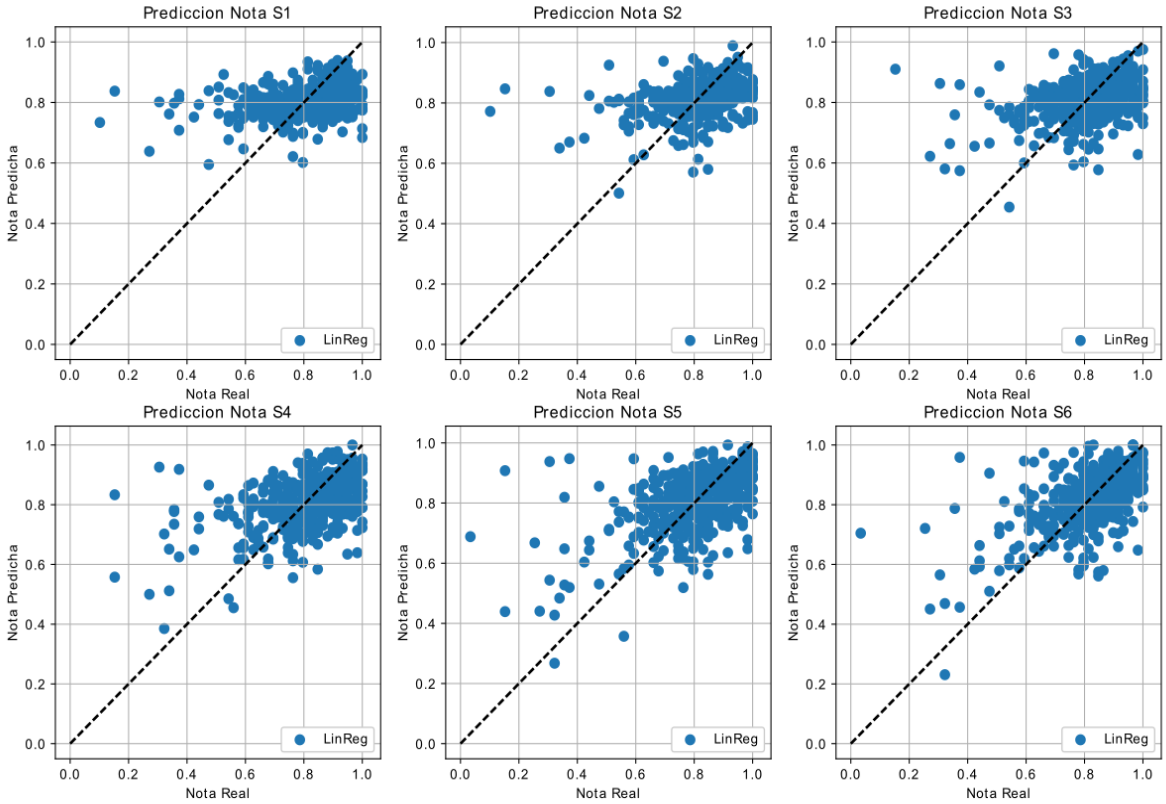

Figura 17. Predicción calificación utilizando características generales en regresión lineal

En las gráficas de la figura 17 podemos observar cómo la predicción de la nota de los estudiantes va mejorando a medida que avanzan las semanas. En esta figura se muestra el algoritmo de regresión lineal aplicado con las características generales que son las que según las tablas nos daban mejores resultados. En el anexo podemos ver el resto de gráficas para cada algoritmo y cada combinación de características. En una situación real solo usaríamos el algoritmo que ha dado mejor resultado junto con las mejores características.

Debido a que los resultados obtenidos no son demasiado buenos analizamos qué ocurriría si usamos la media de los datos de entrenamiento para predecir la nota del examen final de todos los estudiantes. En la tabla 24 se muestra el resultado en error absoluto medio de esta estrategia. Se puede observar que el modelo propuesto mejora la predicción con respecto a predecir siempre la media.

	Resultados con Media	DesvTípica
Semana 1	0,097	0,027
Semana 2	0,091	0,026
Semana 3	0,095	0,025
Semana 4	0,1	0,026
Semana 5	0,101	0,03
Semana 6	0,096	0,028

Tabla 24. Resultados calificación utilizando predicción de la media

En la columna Resultados con Media se muestra el *error absoluto medio* para cada semana si predecimos tomando la media como nota predicha. Si observamos la tabla 22, que es la que nos da los mejores resultados, podemos comprobar que la utilización de características generales y los algoritmos de regresión mejoran los resultados.

5.4.1 Predicción calificación edición 2 a partir de la edición 1

A continuación mostraremos los resultados para la predicción de la nota en la siguiente edición a partir de modelos entrenados usando los datos de la edición 1. El método utilizado es similar al explicado anteriormente. Al no aplicar validación cruzada no aparece una columna con la desviación típica en las tablas.

Problemas			
	LinReg	RFR	XGB
Semana 1	0,197	0,171	0,189
Semana 2	0,203	0,179	0,177
Semana 3	0,225	0,161	0,163
Semana 4	0,242	0,161	0,168
Semana 5	0,182	0,169	0,166
Semana 6	0,196	0,177	0,174

Tabla 25. Resultados calificación utilizando características de problemas, edición 2

Características generales			
	LingReg	RFR	XGB
Semana 1	0,167	0,174	0,178
Semana 2	0,166	0,173	0,172
Semana 3	0,151	0,163	0,168
Semana 4	0,305	0,166	0,167
Semana 5	0,149	0,163	0,162
Semana 6	0,151	0,163	0,163

Tabla 26. Resultados calificación utilizando las características generales, edición 2

Problemas + Características generales			
	LinReg	RFR	XGB
Semana 1	0,2	0,168	0,183
Semana 2	0,222	0,175	0,179
Semana 3	0,215	0,162	0,161
Semana 4	0,279	0,168	0,169
Semana 5	0,183	0,161	0,156
Semana 6	0,17	0,168	0,161

Tabla 27. Resultados calificación utilizando características generales y problemas, edición 2

Tras analizar las tablas 25, 26 y 27 podemos observar que los *errores absolutos* empeoran respecto a la edición anterior debido a que se entrena con la primera edición. Ambas ediciones pueden tener ciertas diferencias como, por ejemplo, el compromiso de los estudiantes con el curso o la utilización del foro. Esto puede generar la diferencia en los errores absolutos de cada edición.

Podemos observar que, en el ámbito de predicción de la nota, las características del resultado de los ejercicios son más relevantes que en el caso de predicción del abandono. Esto se debe principalmente a que el examen final está formado por problemas y por lo tanto saber si un estudiante hace correctamente los problemas puede proporcionar más información para predecir la nota del examen final que para predecir el abandono.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En este trabajo de final de máster se ha creado un sistema que permite analizar distintas ediciones de cursos online de una manera flexible y automática. Para llevar a cabo esta tarea han sido necesarias tres etapas bien diferenciadas: preprocesado de datos con Spark, automatización del flujo de datos y aprendizaje automático.

El preprocesado de los datos se ha realizado mediante la herramienta de manejo de grandes volúmenes de datos Apache Spark. La utilización de dicha tecnología reduce considerablemente el tiempo de ejecución y, por lo tanto, supone una gran ventaja al tratar con grandes cantidades de datos.

En este preprocesado era esencial que el programa funcionase para cualquier curso online de la plataforma edX. Por lo tanto, se ha llevado a cabo una tarea exhaustiva de análisis de posibles eventos realizables por los estudiantes en los cursos online. Estos eventos entre los diferentes cursos tenían que coincidir para poder comparar de los resultados obtenidos más adelante. Tras este preprocesado de datos se obtiene para cada estudiante su información temporal de eventos realizados.

Con la finalidad de poder extraer numerosas características de los cursos online de la manera más rápida y eficiente posible, se ha llevado a cabo la automatización del flujo de los datos extraídos en el preprocesado. Esta automatización consiste en un programa flexible para cualquier curso online y con la posibilidad de añadir y quitar funciones de manera que no se lleve a cabo una modificación del programa en su totalidad. Dentro de las características extraídas en esta automatización se encuentran la información de los problemas resueltos por los estudiantes y la contabilización de eventos de un tipo determinado como por ejemplo el número de interacciones con los videos o número de problemas correctos.

En la última etapa se ha desarrollado un programa que aplica técnicas de aprendizaje automático para la predicción de la calificación y el abandono en cualquier curso online. A partir de este programa hemos podido analizar a los estudiantes en diferentes ediciones de un curso online.

En el caso de abandono se ha observado que para un curso concreto y diferentes criterios podemos establecer un modelo estable a la hora de predecir el abandono en diferentes ediciones del curso online analizado. Si consideramos que los estudiantes abandonan si no realizan eventos en la semana siguiente a la que se encuentra el curso, hemos comprobado que llegamos a predecir con un modelo que posee un *0.8* de área bajo la curva. Si por el contrario consideramos que los estudiantes abandonan si no realizan actividades evaluables las siguientes dos semanas, hemos llegado a obtener un modelo con *0.93* de área bajo la curva. Esta diferencia en los resultados se debe a que el primer criterio es más restrictivo ya que puede ser que haya estudiantes inactivos durante una semana del curso y luego vuelvan a él.

En el caso de la calificación los resultados varían entre distintas ediciones. Esto puede deberse a diversos motivos que pueden ocasionar que las ediciones sean distintas, por ejemplo, uno de esos motivos podría ser la motivación de los estudiantes.

En definitiva, todo lo expuesto anteriormente demuestra que hemos conseguido llevar a cabo un seguimiento de los estudiantes de manera casi automática. Este proceso de seguimiento de los estudiantes es complejo en los cursos online pero mediante el sistema expuesto en el trabajo se puede extraer información relevante del curso permitiendo crear nuevas características de predicción y modelos así como añadir nuevos eventos.

6.2 Trabajo Futuro

Los resultados obtenidos tanto en el abandono como en la predicción de la nota son resultados basados en algunas características sencillas como contar el número de eventos de cada uno de los tipos. Como trabajo posterior se podrían llevar a cabo la creación de nuevas características ya que disponemos de un programa que nos permite la integración de ellas de manera sencilla.

Con la finalidad de poder comparar cursos online diferentes se podrían crear características que no dependan del curso online analizado, por ejemplo, el porcentaje de problemas respondidos o de eventos de algún tipo realizados esa semana. Esto nos permite comparar distintos cursos online y poder establecer un modelo general de predicción de variables de interés para el profesorado.

En otras líneas de trabajo futuro en próximas ediciones del curso online analizado podría aplicarse este sistema para ayudar a los estudiantes del curso y así mejorar su experiencia en él. Además, con la finalidad de hacer nuestro sistema más sencillo de utilizar por profesores que no dispongan de conocimientos informáticos, se podría crear un programa de visualización para facilitarles la tarea del seguimiento de los estudiantes.

Para mejorar los resultados obtenidos en el ámbito del aprendizaje automático se pueden probar nuevos algoritmos, métricas, métodos de reducción de dimensionalidad, extraer la importancia de las variables de predicción etc.

Otra de las ventajas de nuestro sistema es que está basado en tres módulos independientes entre sí (Preprocesado de datos, Flujo de datos y Aprendizaje automático). Por lo tanto se podrían explorar nuevas plataformas educativas (Coursera, Udacity…) y extraer los eventos de los logs de dichas plataformas con la finalidad de pasarlas al programa del flujo de datos en el mismo formato que el expuesto en el trabajo. De esta manera se podrían analizar prácticamente todos los cursos online de las plataformas educativas modificando únicamente el primer módulo.

Referencias

[1] «MOOC» https://es.wikipedia.org/wiki/Mooc [Accedido: 21-enero-2018]

[2] «edX » https://www.edx.org/ [Accedido: 21-enero-2018]

[3] Khalil, M. & Ebner, M. Clustering patterns of engagement in Massive Open Online Courses (MOOCs): the use of learning analytics to reveal student categories. *Journal of Computing in Higher Education* (2017), 29(1), 114-132.

[4] Clark, D. (2016). Moocs: Course completition is the wrong measure of course success. Class Central. https://www.classcentral.com/report/moocs-course-completion-wrong-measure/ [Accedido: 31-enero-2018]

[5] Ramesh, A., Goldwasser, D., Huang, B., Daume III, H., & Getoor, L. Learning latent engagement patterns of students in online courses. *In Twenty-Eighth AAAI Conference on Artificial Intelligence*. (2014).

[6] Yang, D., Sinha, T., Adamson, D., and Rose, C. P. Turn on, tune in, dropout: Anticipating student dropouts in massive open online courses. *Proceedings of the 2013 NIPS Data-driven education workshop 11* (2013), 14. https://www.cs.cmu.edu/~diyiy/docs/nips13.pdf.

[7] Kovanović, V., Joksimović, S., Gašević, D., Owers, J., Scott, A. M., & Woodgate, A.. Profiling MOOC Course Returners: How Does Student Behavior Change Between Two Course Enrollments?. In Proceedings of the Third (2016). *ACM Conference on Learning@ Scale* (pp. 269-272). ACM.

[8] Skryabin, M. Types of Dropout in Adaptive Open Online Courses. *In European Conference on Massive Open Online Courses* (2017) (pp. 273-279). Springer, Cham.

[9] Zhao Z., Wu Q., Chen H. & Wan C. Learning Quality Evaluation of MOOC Based on Big Data Analysis. *In: Qiu M. (eds) Smart Computing and Communication. SmartCom 2016*. Lecture Notes in Computer Science, vol 10135. Springer, Cham.

[10] Hung, J. L., Wang, M., Wang, S., Abdelrasoul, M., & He, W. Identifying at-risk students for early interventions? A time-series clustering approach. *IEEE Transactions on Emerging Topics in Computing*. (2015)

[11]Amnueypornsakul, B., Bhat, S., & Chinprutthiwong, P.. Predicting attrition along the way: the UIUC model. *In Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs* (2014) (pp. 55-59)

[12] Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. Predicting MOOC dropout over weeks using machine learning methods. *In Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs* (2014) (pp. 60-65).

[13] Sharkey, M., & Sanders, R.. A process for predicting MOOC attrition. *In Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs* (2014) (pp. 50-54).

[14] Taylor, C., Veeramachaneni, K., & O'Reilly, U. M. Likely to stop? Predicting stopout in massive open online courses. (2014). arXiv preprint arXiv:1408.3382

[15] Sinha, T., Li, N., Jermann, P., & Dillenbourg, P. Capturing attrition intensifying structural traits from didactic interaction sequences of MOOC learners. (2014). arXiv preprint arXiv:1409.5887.

[16] Boyer, S., & Veeramachaneni, K.. Transfer learning for predictive models in massive open online courses. *In International Conference on Artificial Intelligence in Education* (2015) (pp. 54-63). Springer, Cham.

[17] Kennedy, G., Coffrin, C., De Barba, P., & Corrin, L. Predicting success: how learners' prior knowledge, skills and activities predict MOOC performance. *In Proceedings of the Fifth International Conference on Learning Analytics And Knowledge* (2015) (pp. 136-140). ACM.

[18] Brinton, C. G., Buccapatnam, S., Chiang, M., and Poor, H. V. Mining mooc clickstreams: On the relationship between learner video-watching behavior and performance. arXiv (2015). https://arxiv.org/abs/1503.06489

[19] Brinton, C. G., Buccapatnam, S., Chiang, M., & Poor, H. V. (2016). Mining MOOC Clickstreams: Video-Watching Behavior vs. In-Video Quiz Performance. *IEEE Transactions on Signal Processing*, 64(14), 3677-3692.

[20] Xu, B., & Yang, D. (2016). Motivation classification and grade prediction for MOOCs learners. *Computational intelligence and neuroscience*. (2016) 4.

[21] Allione, G., and Stein, R. M. Mass attrition: An analysis of drop out from principles of microeconomics mooc. *The Journal of Economic Education* 47, 2 (2016), 174-186. http://www.tandfonline.com/doi/abs/10.1080/00220485.2016.1146096?journalCode=vece20&

[22] Reischer, M., Khalil, M., & Ebner, M. Does Gamification in MOOC Discusion Forums Work? *In Digital Education: Out to the World and Back to the Campus* (2017) (pp. 95-101). (Lecture Notes in Computer Science; Vol. 10254). Springer International Publishing AG . DOI: 10.1007/978-3-319-59044-8_11

[23] Wen, M., Yang, D., and Rose, C. P. Sentiment analysis in mooc discussion forums: What does it tell us. *Proceedings of educational data mining 1* (2014). http://www.cs.cmu.edu/~mwen/papers/edm2014-camera-ready.pdf.

[24] Er, E., Gómez-Sánchez, E., Bote-Lorenzo, M.L., Dimitriadis, Y. & Asensio-Pérez, J.I. Predicting Peer-Review Participation at Large Scale Using an Ensemble Learning Method. *Proceedings of the Learning Analytics Summer Institute Spain*. 2017, Madrid, Spain, July 2017.

[25] Costello, E., Brown, M., Nair, B., Nic, M., Mhichíl, G., Zhang, J. & Lynn, T. #MOOC Friends and Followers: *An Analysis of Twitter Hashtag Networks*, (2017) DOI 10.1007/978-3-319-59044-8_19.

[26] «Spark» http://spark.apache.org/ [Accedido: 21-enero-2018]

[27] «Components Spark» http://spark.apache.org/docs/latest/cluster-overview.html [Accedido: 21-enero-2018]

[28] «MapReduce Tutorial» https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html [Accedido: 21-enero-2018]

[29] Breiman, L.: Random forests. *Machine learning* 45(1), 5–32 (2001)

[30] «XGBoost» https://arxiv.org/pdf/1603.02754.pdf [Accedido: 6-febrero-2018]

Anexo

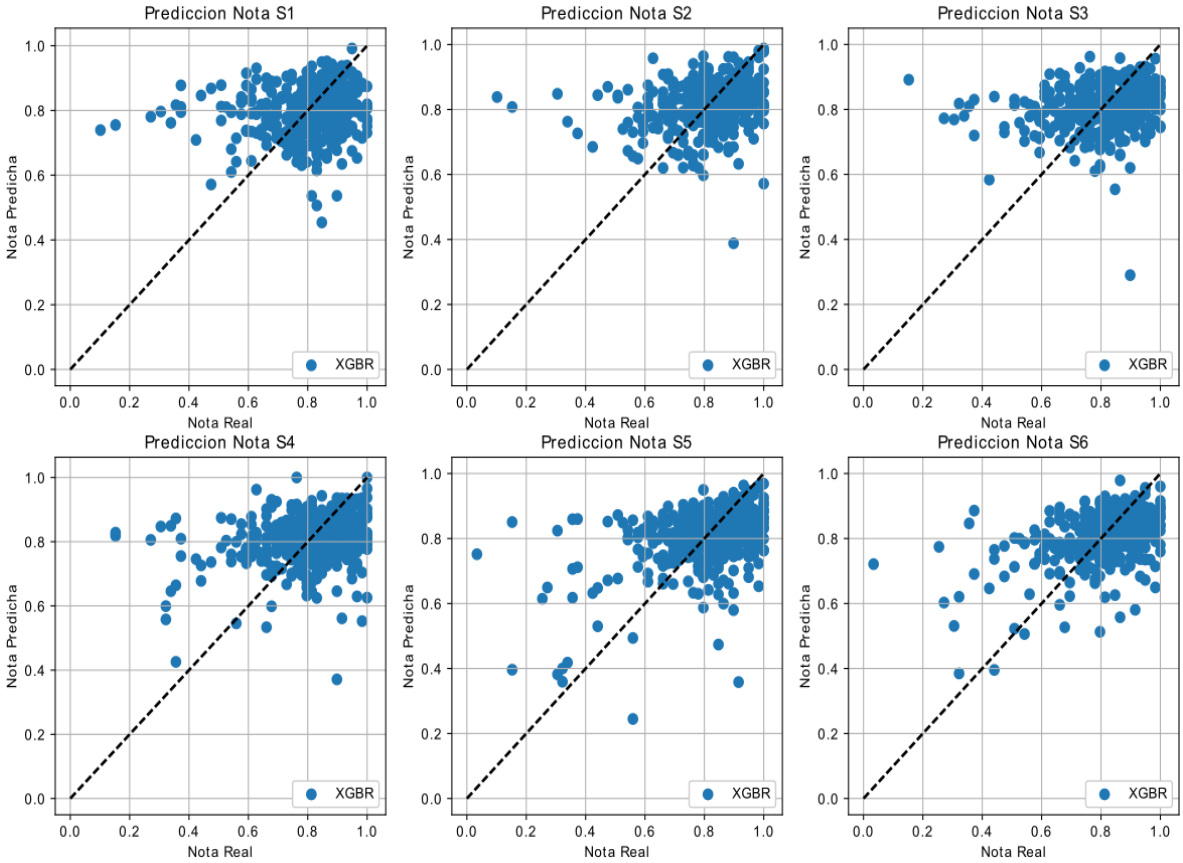

Figura 18. Predicción calificación utilizando características generales en xgboost.

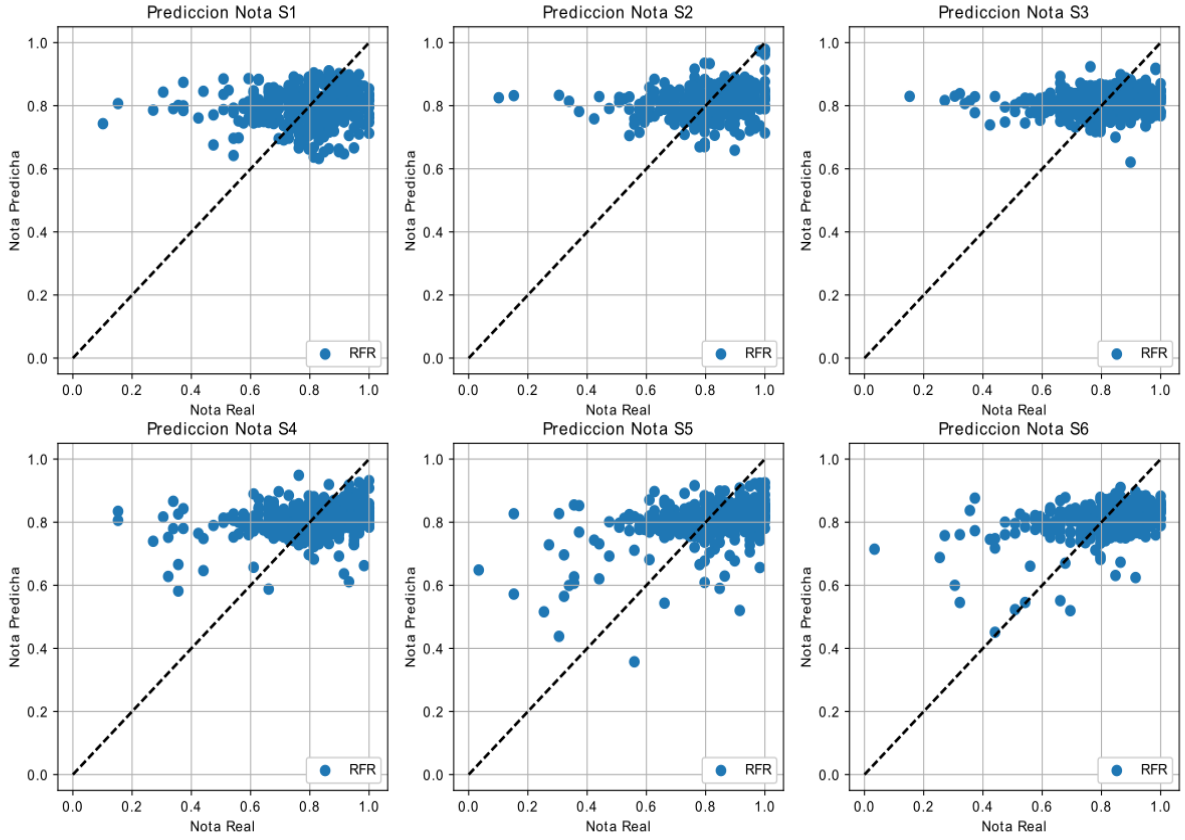

Figura 19. Predicción calificación utilizando características generales en random forest.

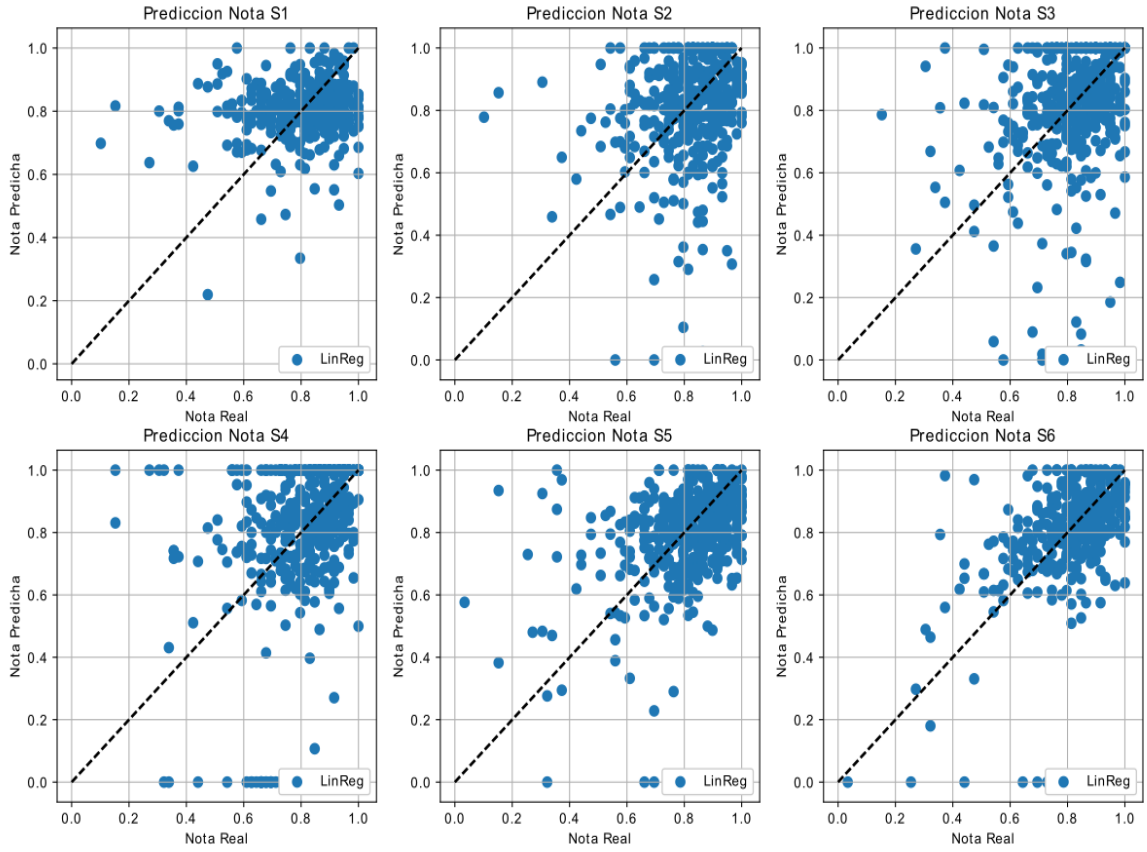

Figura 20. Predicción calificación utilizando características generales y de problemas en regresión lineal.

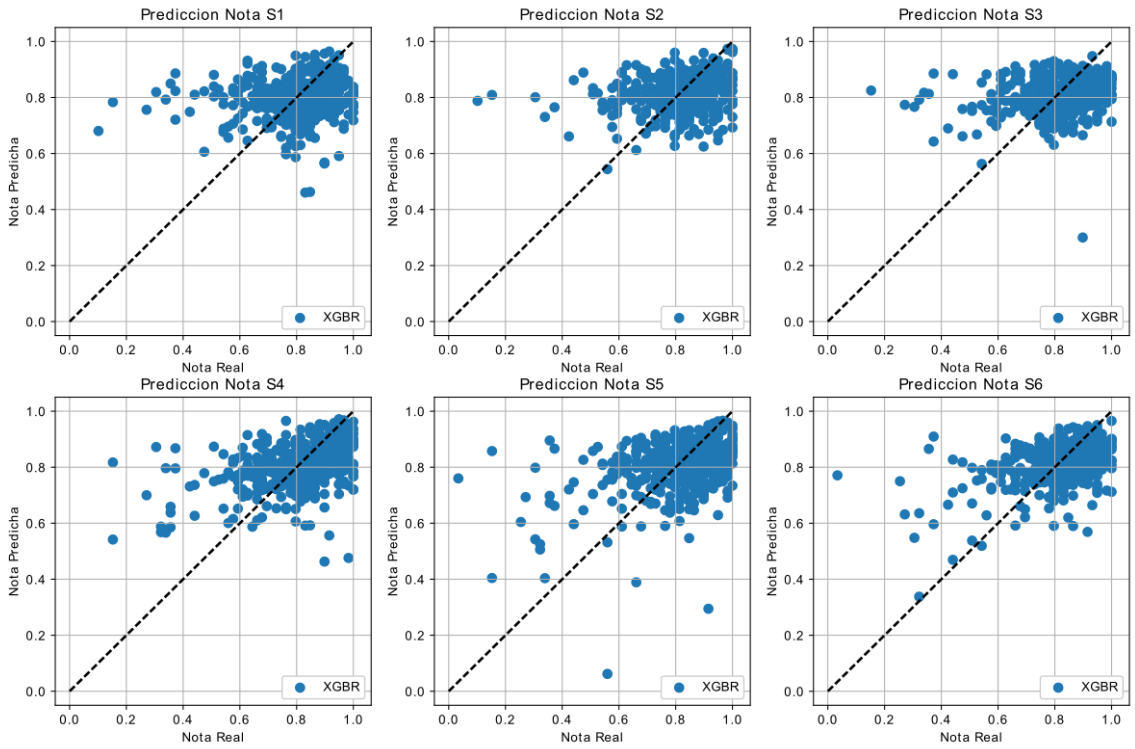

Figura 21. Predicción calificación utilizando características generales y de problemas en xgboost.

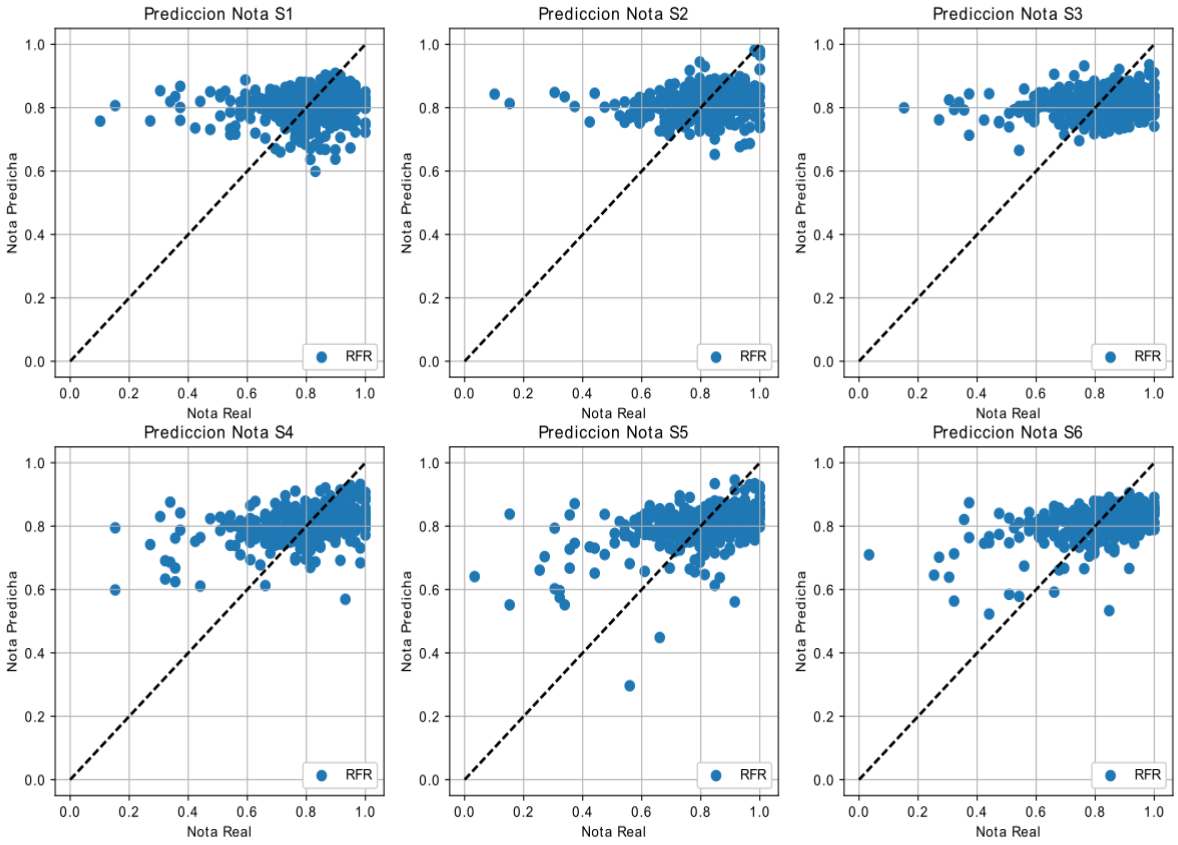

Figura 22. Predicción calificación utilizando características generales y de problemas en random forest.

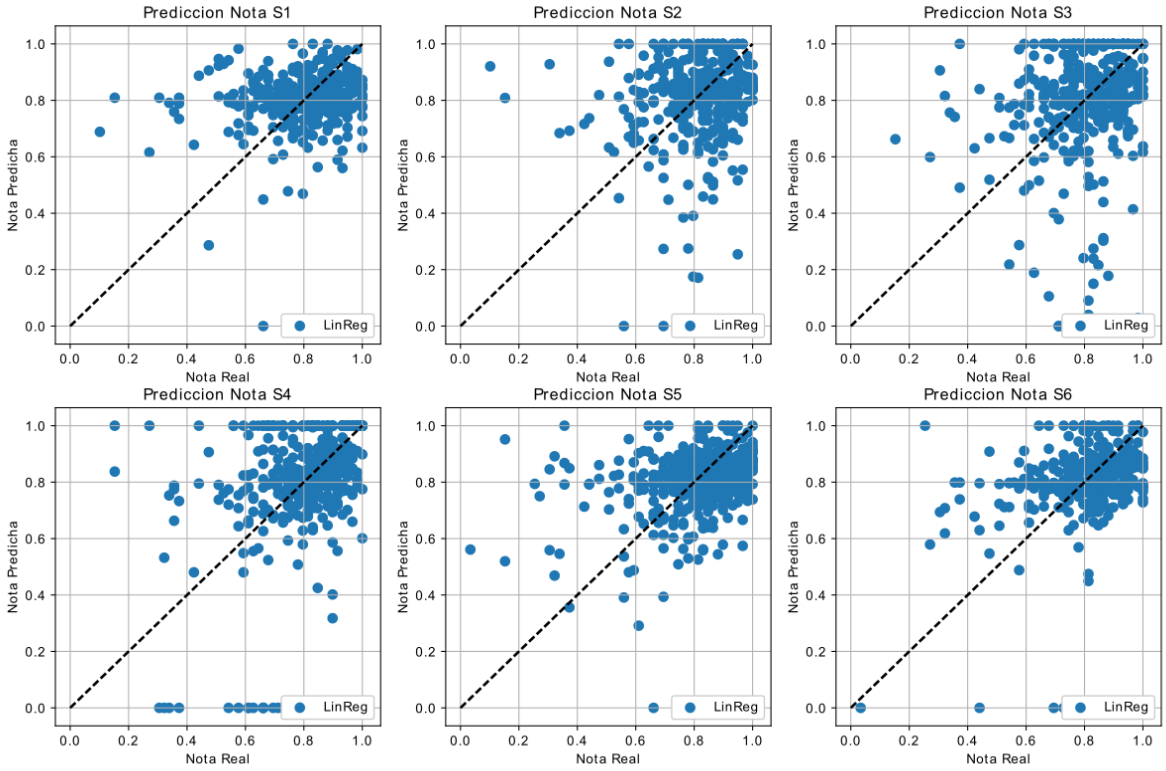

Figura 23. Predicción calificación utilizando características de problemas en regresión lineal.

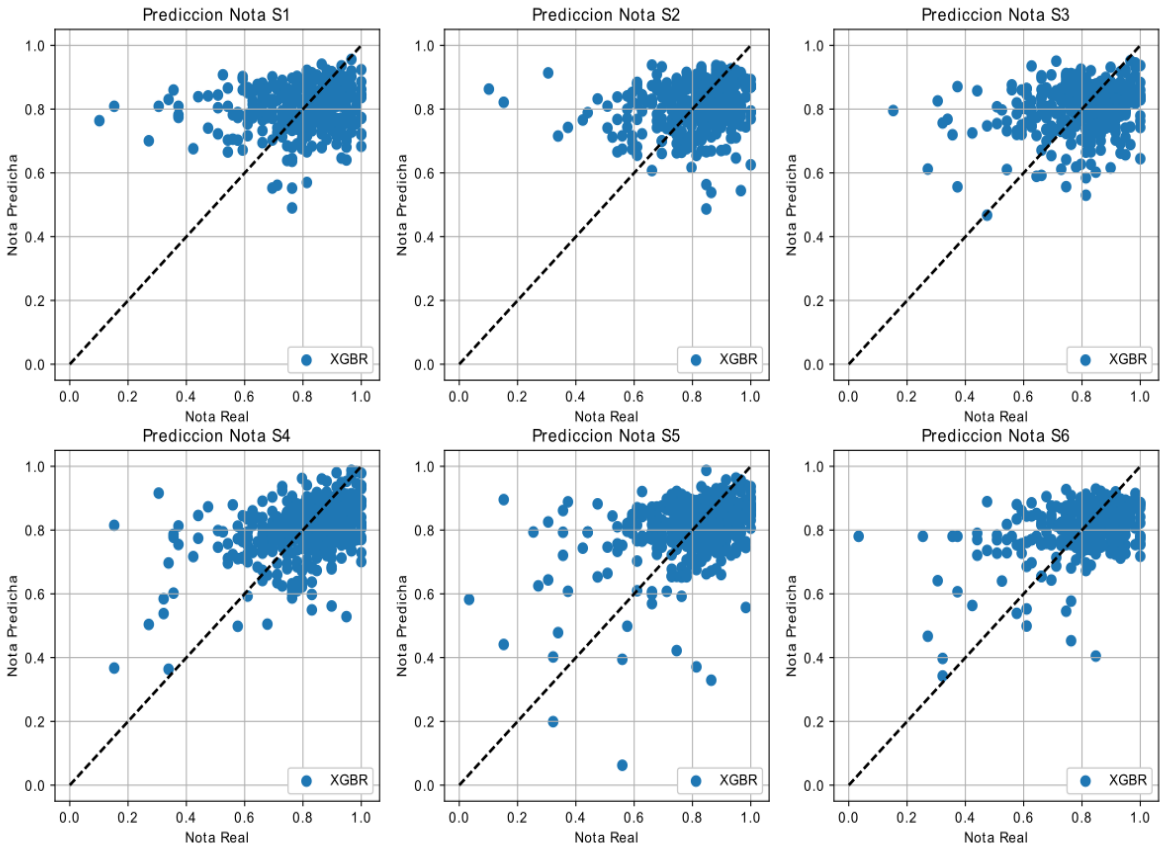

Figura 24. Predicción calificación utilizando características de problemas en xgboost.

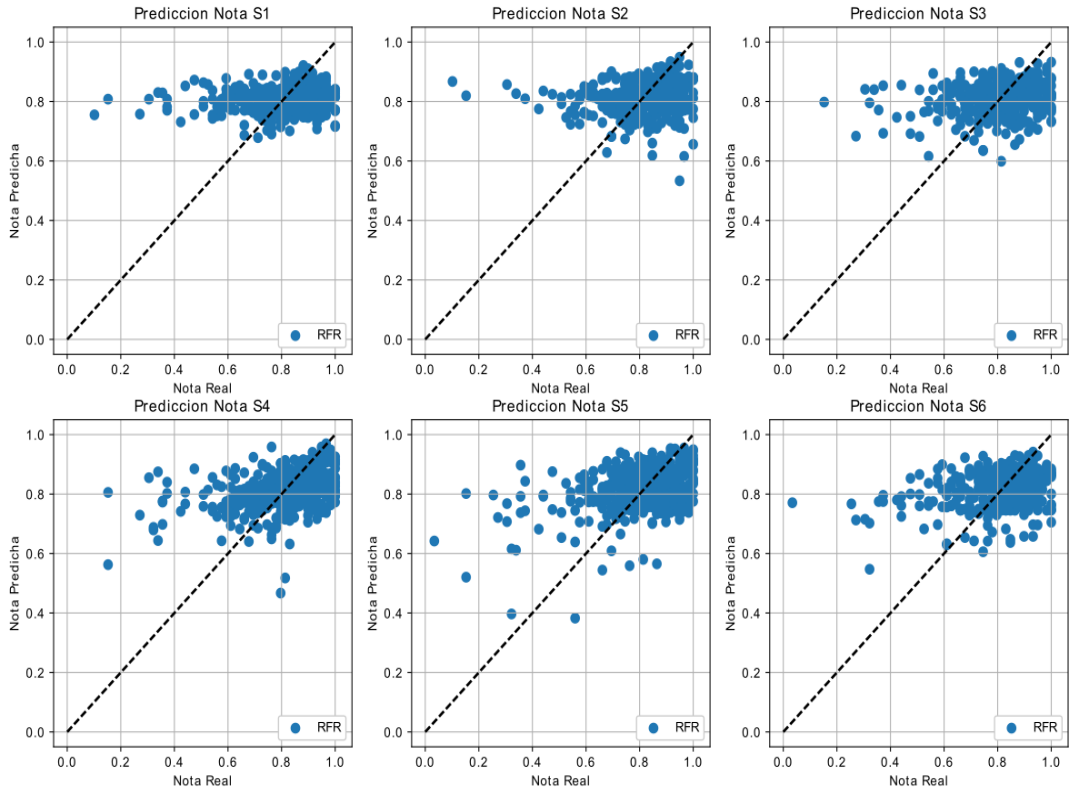

Figura 25. Predicción calificación utilizando características de problemas en random forest.